Access DB# 112927

(#100)

SEARCH REQUEST FORM

Scientific and Technical Information Center

Requester's Full Name: Qamrun Nahar Examiner #: 79621 Date: 1/28/04
Art Unit: 2124 Phone Number 305-7699 Serial Number: 09/747,224
Mail Box and Bldg/Room Location: PK2-5B46 Results Format Preferred (circle): PAPER DISK E-MAIL

If more than one search is submitted, please prioritize searches in order of need.

****

Please provide a detailed statement of the search topic, and describe as specifically as possible the subject matter to be searched. Include the elected species or structures, keywords, synonyms, acronyms, and registry numbers, and combine with the concept or utility of the invention. Define any terms that may have a special meaning. Give examples or relevant citations, authors, etc, if known. Please attach a copy of the cover sheet, pertinent claims, and abstract.

Title of Invention: Assembly language code compilation for an instruction-set architecture containing new instructions using the prior assembler

Inventors (please provide full names): John Simons

Earliest Priority Filing Date: 12/22/00

***For Sequence Searches Only* Please include all pertinent information (parent, child, divisional, or issued patent numbers) along with the appropriate serial number.**

Keyword: cross-assembler

Uses an old or prior assembler to assemble source code containing both old and new assembly instructions of a new instruction-set architecture to produce corresponding object code.
The source code is first examined by a preprocessor that writes old instruction to a temporary file unchanged. when a new instruction is encountered, converts the new instruction to ASCII OP code equivalent and then write ASCII OP code to temporary file as data directive. Then, the temp file is applied to the old Assembler to produce a machine language program executable by the new instruction-set simulator, or the new processor, if available.

****

STAFF USE ONLY	Type of Search		Vendors and cost where applicable
Searcher: Geoffrey St. Lege	NA Sequence (#)		STN
Searcher Phone #: 308-7800	AA Sequence (#)		Dialog ✓
Searcher Location: 4B30	Structure (#)		Questel/Orbit
Date Searcher Picked Up: 1/28/4	Bibliographic	✓	Dr.Link
Date Completed: 2/12/4	Litigation		Lexis/Nexis
Searcher Prep & Review Time: 1/28/04	Fulltext	✓	Sequence Systems
Clerical Prep Time: 95	Patent Family		WWW/Internet
Online Time: 250	Other		Other (specify)

PTO-1590 (8-01)

STIC Search Report
EIC 2100

STIC Database Tracking Number: 112927

TO:	From:
Qamrun Nahar Location: Art Unit : 2124 Monday, February 02, 2004 Case Serial Number: 09747824	Geoffrey St. Leger Location: EIC 2100 PK2-4B30 Phone: 308-7800 geoffrey.stleger@uspto.gov

Search Notes

Dear Examiner Nahar,

Attached please find the results of your search request for application 09747824. I searched Dialog's foreign patent files, technical databases, product announcement files and general files.

Please let me know if you have any questions.

Regards,

Geoffrey St. Leger
4B30/308-7800

File 347:JAPIO Oct 1976-2003/Sep(Updated 040105)
(c) 2004 JPO & JAPIO
File 350:Derwent WPIX 1963-2004/UD,UM &UP=200407
(c) 2004 Thomson Derwent
File 348:EUROPEAN PATENTS 1978-2004/Jan W05
(c) 2004 European Patent Office
File 349:PCT FULLTEXT 1979-2002/UB=20040129,UT=20040122
(c) 2004 WIPO/Univentio

Set	Items	Description
S1	176	AU=SIMONS J?
S2	23	S1 AND ASSEMBL?
S3	3	S2 AND IC=G06F

3/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO

07328449 **Image available**
METHOD FOR COMPILING **ASSEMBLY** LANGUAGE CODE FOR INSTRUCTION SET ARCHITECTURE INCLUDING NEW INSTRUCTION USING CONVENTIONAL **ASSEMBLER**

PUB. NO.: 2002-196937 [JP 2002196937 A]
PUBLISHED: July 12, 2002 (20020712)
INVENTOR(s): **SIMONS JOHN**
APPLICANT(s): HITACHI LTD
APPL. NO.: 2001-328956 [JP 20011328956]
FILED: October 26, 2001 (20011026)
PRIORITY: 00 747824 [US 2000747824], US (United States of America), December 22, 2000 (20001222)
INTL CLASS: **G06F-009/45**

ABSTRACT

PROBLEM TO BE SOLVED: To provide a simple and effective method for compiling an instruction of a new **assembly** language instruction set architecture using a previous **assembler** .

SOLUTION: An **assembler** expansion instruction set architecture ISA is formed from the newest ISA to which a new instruction is added. The **assembly** of a source code output in hybrid of present and new **assembly** language instructions is attained by preprocessing of the source code in order to generate a temporary file 46 including a previous instruction 40a and data designation 41 to each of new **assembly** instructions having object code equivalence of the new instruction 40b as a data independent function, Thereafter, the temporary file 46 is used for the previous **assembler** 40a in order to generate object codes corresponding to each of the previous **assembly** language instructions. Resultantly, an executable machine language program for the new ISA is generated after linking.

3/5/2 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX

015692813 **Image available**
WPI Acc No: 2003-755002/200371
XRPX Acc No: N03-604956
Source code conversion in e.g. microcomputer, involves applying instructions of temporary file to assembler to produce object code corresponding to old instructions, and data forming object code for new instructions
Parent Assignee: HITACHI LTD (HITA); HITACHI AMERICA LTD (HITA)
Inventor: **SIMONS J**
Number of Countries: 002 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20020083421	A1	20020627	US 2000747824	A	20001222	200371	B
JP 2002196937	A	20020712	JP 2001328956	A	20011026	200371	

Priority Applications (No Type Date): US 2000747824 A 20001222
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20020083421	A1		8	G06F-009/45	
JP 2002196937	A		7	G06F-009/45	

Abstract (Basic): US 20020083421 A1
NOVELTY - The method involves copying several instructions to a temporary file (46). The instructions of the temporary file are applied to an **assembler** to produce an object code corresponding to the old instructions (40a,40b), and a data forming object code for new instructions.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for a source code **assembling** method.

USE - For converting source code to object code in microcomputers and microprocessors.

ADVANTAGE - Enables **assembling** code for new instruction-set architecture (ISA) using older **assembler** such that development of extended ISA can continue concomitant with the development of new **assembler** .

DESCRIPTION OF DRAWING(S) - The figure shows a schematic view explaining source code conversion.

source file (40)

old **assembly** instructions (40a,40b)

preprocessor (42)

temporary source code file (46)

object file (50)

new processor (58)

pp; 8 DwgNo 3/4

File Terms: SOURCE; CODE; CONVERT; MICROCOMPUTER; APPLY; INSTRUCTION; TEMPORARY; FILE; **ASSEMBLE** ; PRODUCE; OBJECT; CODE; CORRESPOND; INSTRUCTION; DATA; FORMING; OBJECT; CODE; NEW; INSTRUCTION

Derwent Class: T01

International Patent Class (Main): **G06F-009/45**

File Segment: EPI

3/5/3 **(Item 2 from file: 350)**

011025664 **Image available**

WPI Acc No: 1997-003588/199701

XRPX Acc No: N97-003171

Automatic debugging command file formation for computer system - by re-assembling corrected source program counter which includes implanting debugging command at new opposed-position in automatically generating new command file with new brake point command

Patent Assignee: HITACHI LTD (HITA); HITACHI AMERICA LTD (HITA)

Inventor: SHRIDHAR A; **SIMONS J**

Number of Countries: 002 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 8272648	A	19961018	JP 95285382	A	19951101	199701	B
US 5815714	A	19980929	US 94366050	A	19941229	199846	
			US 97839229	A	19970421		

Priority Applications (No Type Date): US 94366050 A 19941229; US 97839229 A 19970421

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 8272648	A		13	G06F-011/28	
US 5815714	A			G06F-009/45	Cont of application US 94366050

Abstract (Basic): JP 8272648 A

The method involves embedding one debugging command to one line of a source program counter. A brake point command is formed corresp. to each extracted implanting debugging command. A command file is generated by writing the brake point command and the debugging command to a debugging command file. An object code is formed from the source program command which is retouched by removing the brake point command and the implanting debugging command from the debugging command file.

A new command file which includes the new brake point command corresp. to the implanting debugging command, is generated. The implanting debugging command is in a new opposed-position on the source program counter during re- **assembling** of the corrected source program counter.

ADVANTAGE - Enables automatic formation of debugging command file utilised by debugger during simulation execution of object code drawn from source program counter.

Dwg.2/8
Title Terms: AUTOMATIC; DEBUG; COMMAND; FILE; FORMATION; COMPUTER; SYSTEM; **ASSEMBLE** ; CORRECT; SOURCE; PROGRAM; COUNTER; IMPLANT; DEBUG; COMMAND; NEW; OPPOSED; POSITION; AUTOMATIC; GENERATE; NEW; COMMAND; FILE; NEW; BRAKE; POINT; COMMAND
Derwent Class: T01
International Patent Class (Main): **G06F-009/45** ; **G06F-011/28**
File Segment: EPI

File 347:JAPIO Oct 1976-2003/Sep(Updated 040105)
(c) 2004 JPO & JAPIO
File 350:Derwent WPIX 1963-2004/UD,UM &UP=200407
(c) 2004 Thomson Derwent

Set	Items	Description
S1	837333	ASSEMBL???
S2	37232	(ASSEMBL? OR COMPIL???)(5N)(CODE OR INSTRUCTION? ? OR PROGRAM OR FILE OR DATA OR INFORMATION OR OPERATION? ? OR OPERATOR? ? OR COMMAND? ? OR FUNCTION? ? OR DIRECTIVE? ? OR PROCEDURE? ?)
S3	1026	INSTRUCTION()SET()ARCHITECTURE? ? OR ISA
S4	10326	(NEW??? OR UPDAT? OR UPGRAD? OR RECENT? OR LATEST OR ADDED OR ADDITIONAL OR SUPPLEMENTARY OR EXTRA)(5N)(INSTRUCTION? ? OR OPERATOR? ? OR COMMAND? ?)
S5	5992	(OLD?? OR PREVIOUS? OR PRIOR OR PRECEDING OR FORMER? OR ORIGINAL OR INITIAL OR PRIMARY)(5W)(INSTRUCTION? ? OR OPERATOR? ? OR COMMAND? ?)
S6	228192	(TRANSFORM? OR TRANSLAT? OR CONVERT??? OR CONVERSION? ? OR CHANG? OR MODIF???? OR MODIFICATION? ? OR AMEND? OR ADJUST??? OR ADJUSTMENT? ? OR ALTER??? OR ALTERATION? ?)(5N)(INSTRUCTION? ? OR OPERATOR? ? OR COMMAND? ? OR CODE OR DATA)
S7	195	S6(5N)(ASCII OR OPCODE OR OP()CODE)
S8	45	CROSS()ASSEMBL??? OR CROSSASSEMBL???
S9	2	S1 AND S7
S10	2	S1 AND S2 AND S3 AND S4:S5
S11	2	S1 AND S3 AND S4:S5
S12	46	S1 AND S3
S13	48	S9:S12
S14	20	S13 AND IC=G06F
S15	287	S1 AND S4:S5
S16	108	S15 AND IC=G06F
S17	71	S16 AND S2
S18	7	S8 AND IC=G06F
S19	38	S8 NOT S18

14/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO

07328449 **Image available**
METHOD FOR **COMPILING** **ASSEMBLY** LANGUAGE CODE FOR **INSTRUCTION** SET **ARCHITECTURE** INCLUDING **NEW** **INSTRUCTION** USING CONVENTIONAL **ASSEMBLER**

PUB. NO.: 2002-196937 [JP 2002196937 A]
PUBLISHED: July 12, 2002 (20020712)
INVENTOR(s): SIMONS JOHN
APPLICANT(s): HITACHI LTD
APPL. NO.: 2001-328956 [JP 20011328956]
FILED: October 26, 2001 (20011026)
PRIORITY: 00 747824 [US 2000747824], US (United States of America), December 22, 2000 (20001222)
INTL CLASS: **G06F-009/45**

ABSTRACT
PROBLEM TO BE SOLVED: To provide a simple and effective method for **compiling** an **instruction** of a new **assembly** language **instruction** **set** **architecture** using a **previous** **assembler** .

SOLUTION: An **assembler** expansion **instruction** **set** **architecture** **ISA** is formed from the **newest** **ISA** to which a **new** **instruction** is **added** . The **assembly** of a source **code** output in hybrid of present and **new** **assembly** language **instructions** is attained by preprocessing of the source code in order to generate a temporary file 46 including a **previous** **instruction** 40a and data designation 41 to each of **new** **assembly** **instructions** having object **code** equivalence of the **new** **instruction** 40b as a data independent function, Thereafter, the temporary file 46 is used for the previous **assembler** 40a in order to generate object codes corresponding to each of the **previous** **assembly** language **instructions** . Resultantly, an executable machine language program for the new **ISA** is generated after linking.

14/5/2 (Item 2 from file: 347)
DIALOG(R)File 347:JAPIO

06037114 **Image available**
COMPILE SYSTEM AND COMPUTER PROGRAM PRODUCT

PUB. NO.: 10-320214 [JP 10320214 A]
PUBLISHED: December 04, 1998 (19981204)
INVENTOR(s): JACK GREENEBAUM
MICHAEL BAXTER
APPLICANT(s): RICOH CO LTD [000674] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 10-097703 [JP 9897703]
FILED: April 09, 1998 (19980409)
PRIORITY: 7-827,619 [US 827619-1997], US (United States of America), April 09, 1997 (19970409)
INTL CLASS: [6] **G06F-009/45**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JAPIO KEYWORD:R131 (INFORMATION PROCESSING -- Microcomputers & Microprocessers)

ABSTRACT
PROBLEM TO BE SOLVED: To dynamically generate an execution file to be used for a resettable processing unit by compiling a source code written in C or pascal and constructing selectively exchangeable internal hardware structure.

SOLUTION: A C compiler 402 reads a source file 401 containing a source code

instruction statement from a disk memory or the like and identifies instruction set structure (**ISA**) concerning the subset of source code instruction statement. Next, an **assembly** language statement is prepared by compiling the identified subset of instruction to be executed by the **ISA** . Then, when the end of source file is reached, the **assembly** language statement is **assembled** by an **assembler** 409 and an object file 403 is prepared. The object file 403 is linked with a bit stream position while using a software linker 404, which can deal with 64-bit arrangement addresses, and an execution file 405 is prepared.

14/5/3 (Item 3 from file: 347)
DIALOG(R)File 347:JAPIO

02481154 **Image available**
INTELLIGENT I/O DEVICE

PUB. NO.: 63-098054 [JP 63098054 A]
PUBLISHED: April 28, 1988 (19880428)
INVENTOR(S): OGASAWARA AKIHIRO
KURIMOTO TAKESHI
APPLICANT(S): TOSHIBA CORP [000307] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 61-242687 [JP 86242687]
FILED: October 13, 1986 (19861013)
INTL CLASS: [4] **G06F-013/12** ; G05B-015/02; H04L-013/00
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- Memory Units); 22.3 (MACHINERY -- Control & Regulation); 44.3 (COMMUNICATION -- Telegraphy)
JOURNAL: Section: P, Section No. 757, Vol. 12, No. 339, Pg. 48, September 12, 1988 (19880912)

ABSTRACT

PURPOSE: To prevent a quantity of data transmission from being increased, and to reduce load on an input/output controller even when an external equipment is **assembled** , by incorporating a CPU in an intelligent I/O device itself, and performing the conversion processing of a transmission format by the device itself.

CONSTITUTION: When a data is transmitted from the intelligent I/O device 12 to the external equipment 5, the CPU reads out a binary data from the specific address of a memory RAM17, and **converts** it to an **ASCII code** , and transmits the coded data to the external equipment 5 through a transmission line 21. When the data is transmitted from the external equipment 5 to the intelligent I/O device 12, and when the intelligent I/O device 12 receives the ASCII code from the external equipment 5 at a transmission/reception part 8, the CPU16, after applying a BCD conversion processing on the data, and converting it to the binary data, stores it in the RAM17.

14/5/4 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX

015692813 **Image available**
WPI Acc No: 2003-755002/200371
XRPX Acc No: N03-604956
Source code conversion in e.g. microcomputer, involves applying instructions **of temporary** file **to** assembler **to produce object** code **corresponding to** old instructions **, and data forming object code for** new instructions
Patent Assignee: HITACHI LTD (HITA); HITACHI AMERICA LTD (HITA)
Inventor: SIMONS J
Number of Countries: 002 Number of Patents: 002
Patent Family:
Patent No Kind Date Applicat No Kind Date Week

US 20020083421 A1 20020627 US 2000747824 A 20001222 200371 B
JP 2002196937 A 20020712 JP 2001328956 A 20011026 200371

Priority Applications (No Type Date): US 2000747824 A 20001222
Patent Details:
Patent No Kind Lan Pg Main IPC Filing Notes
US 20020083421 A1 8 G06F-009/45
JP 2002196937 A 7 G06F-009/45
Abstract (Basic): US 20020083421 A1

NOVELTY - The method involves copying several instructions to a temporary file (46). The instructions of the temporary **file** are applied to an **assembler** to produce an object **code** corresponding to the **old instructions** (40a,40b), and a data forming object code for **new instructions** .

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for a source **code assembling** method.

USE - For converting source code to object code in microcomputers and microprocessors.

ADVANTAGE - Enables **assembling code** for **new instruction - set architecture** (**ISA**) using older **assembler** such that development of extended **ISA** can continue concomitant with the development of new **assembler** .

DESCRIPTION OF DRAWING(S) - The figure shows a schematic view explaining source code conversion.

source file (40)
old assembly instructions (40a,40b)
preprocessor (42)
temporary source code file (46)
object file (50)
new processor (58)
pp; 8 DwgNo 3/4

Title Terms: SOURCE; CODE; CONVERT; MICROCOMPUTER; APPLY; INSTRUCTION; TEMPORARY; FILE; **ASSEMBLE** ; PRODUCE; OBJECT; CODE; CORRESPOND; INSTRUCTION; DATA; FORMING; OBJECT; CODE; NEW; INSTRUCTION
Derwent Class: T01
International Patent Class (Main): **G06F-009/45**
File Segment: EPI

17/5/5 (Item 5 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

06233781 **Image available**
EXTENSION INSTRUCTION SET EMULATOR

PUB. NO.: 11-175352 [JP 11175352 A]
PUBLISHED: July 02, 1999 (19990702)
INVENTOR(s): YASUDA MITSUHIRO
APPLICANT(s): MITSUBISHI ELECTRIC CORP
APPL. NO.: 09-346016 [JP 97346016]
FILED: December 16, 1997 (19971216)
INTL CLASS: **G06F-009/455**

ABSTRACT

PROBLEM TO BE SOLVED: To comply with various user's requests that user have freely set by providing an additional processing program part which is placed before or behind a simulation part and a core-incorporated **program** part of an **assembler instruction** and described at a different level from the **assembler** level and can directly be simulated by a simulated target computer.

SOLUTION: This simulator has a program counter 27, an on-instruction execution part 25, a command control part 39, etc., as functions that a conventional **instruction** set simulator has and is **newly** provided with a preprocessing part 24 that a user can defines in front of the one-instruction execution part 25 and a postprocessing part 26 that the user can define behind the execution part 25. Thus, the additional processing program part is provided which is placed before or behind the simulation part and core-incorporated **program** part of an **assembler instruction** consisting of only an **instruction** set that the target computer itself has and described at the different level from the **assembler** level and can directly be simulated by the simulated target computer.

17/5/6 (Item 6 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

05728189 **Image available**
INSTRUCTION NUMBER EXPANSION METHOD IN PARALLEL PROCESSOR, AND PARALLEL PROCESSORS

PUB. NO.: 10-011289 [JP 10011289 A]
PUBLISHED: January 16, 1998 (19980116)
INVENTOR(s): KONDO YOSHIKAZU
APPLICANT(s): MITSUBISHI ELECTRIC CORP [000601] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 08-158505 [JP 96158505]
FILED: June 19, 1996 (19960619)
INTL CLASS: [6] **G06F-009/38** ; **G06F-009/30**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)

ABSTRACT

PROBLEM TO BE SOLVED: To practically reduce instruction extension, to mitigate the limit of an I/O band width and to increase functions mountable to one substrate by respectively writing plural control codes corresponding to plural instructions to plural instruction decoders based on decoding **information** prepared at the time of **assembling** a source **file** .
SOLUTION: An **instruction code** successive allocation **assembler** 4 outputs which **instruction** has been allocated to which instruction, to an instruction name-instruction code correspondence table 300 for indicating the correspondence relation of an instruction name and the instruction code. A control code generator 5 finds the control code corresponding to

the instruction code based on the instruction name-instruction code correspondence table 300 and a virtual instruction set 400 for indicating the correspondence relation of the instructions and all the control codes physically realizable on the parallel processor 600 and outputs it to a decoding memory table 500. Then, the contents of the decoding memory table 500 are transferred to a decoding memory group (instruction decoder) **prior** to the activation of the **instruction** .

17/5/7 (Item 7 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

05279498 **Image available**
ASSEMBLER PROCESSING METHOD

PUB. NO.: 08-234998 [JP 8234998 A]
PUBLISHED: September 13, 1996 (19960913)
INVENTOR(s): SHAMOTO EIJI
APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 07-040214 [JP 9540214]
FILED: February 28, 1995 (19950228)
INTL CLASS: [6] **G06F-009/45**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)

ABSTRACT

PURPOSE: To provide an **assembler** processing method which can substantially shorten the syntax analysis processing time.

CONSTITUTION: This processing method includes a syntax analysis part 105 which generates the reference information for every symbol reference, generates the optimization information for every optimization instruction and produces an intermediate code from the input data, an optimization processing part 111 which optimizes the optimization instructions, and a code generation processing part 112 which solves the undecided operand included in the intermediate code. Furthermore, a means is **added** to fast perform an **instruction** optimization **assembling operation** .

17/5/9 (Item 9 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

04854196 **Image available**
OPTIMIZING DEVICE FOR PROGRAM AND OPTIMIZING METHOD FOR PROGRAM

PUB. NO.: 07-146796 [JP 7146796 A]
PUBLISHED: June 06, 1995 (19950606)
INVENTOR(s): HASHIGUCHI WATARU
APPLICANT(s): MATSUSHITA ELECTRIC IND CO LTD [000582] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 06-127229 [JP 94127229]
FILED: June 09, 1994 (19940609)
INTL CLASS: [6] **G06F-009/45**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)

ABSTRACT

PURPOSE: To easily detect an unnecessary instruction having a content which is substantially overlapped with a **previous instruction** at the time of optimizing a program, to improve the execution speed of the program and to compress a program size.

CONSTITUTION: A register value extraction circuit 12 extracts the value of a register, which is virtually executed and decided, as the internal state of a computer and stores it in a storage device 104 for the respective **instructions** in an **assembler program** . When the value of the register, which is extracted for one instruction, is at least a part of the value of

the register, which is extracted on the **previous instruction** , an unnecessary **instruction** decision circuit 105 decides that the subsequent instruction is the unnecessary instruction which is substantially overlapped with the **previous instruction** . A correction circuit 106 eliminates the unnecessary instruction from the program. The unnecessary instruction can be left in the program and the execution can be inhibited. Thus, the unnecessary instruction in the program can easily be detected.

17/5/11 (Item 11 from file: 347)
DIALOG(R)File 347:JAPIO

04213003 **Image available**
ASSEMBLY INSTRUCTION DISPLAY SYSTEM

PUB. NO.: 05-204703 [JP 5204703 A]
PUBLISHED: August 13, 1993 (19930813)
INVENTOR(s): UEDA KUMIKO
APPLICANT(s): NEC IC MICROCOMPUT SYST LTD [470861] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 04-013568 [JP 9213568]
FILED: January 29, 1992 (19920129)
INTL CLASS: [5] **G06F-011/28** ; **G06F-009/06**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 1649, Vol. 17, No. 634, Pg. 73, November 24, 1993 (19931124)

ABSTRACT

PURPOSE: To facilitate debugging in the case of debugging and to facilitate bug correction by displaying table numbers showing the order of **assembly** descriptions in an **assemble** list after scheduling **instructions** .

CONSTITUTION: A source program **file** 1 is processed through an **assembler** 10 and outputted to an **assemble** list **file** 2 and an object module **file** 3. The **assembler** 10 executes processings 11, 12 and 14-17 by using an **assembly instruction information** table 14. Namely, line number **information** is **added** to **assembly instructions** before the scheduling processing 15 by an **assemble instruction information** table preparation processing 12. With the addition of instructions, line numbers are generated by the instruction scheduling processing 15. Then, in the display processing 17 for line numbers and **assembly instructions** after the **instruction** scheduling processing, are displayed the line numbers showing the order of **assembly** descriptions in the **assemble** list after the **instruction** scheduling processing 15 by these two means.

17/5/12 (Item 12 from file: 347)
DIALOG(R)File 347:JAPIO

04189964 **Image available**
PROGRAM PATCH CORRECTING SYSTEM

PUB. NO.: 05-181664 [JP 5181664 A]
PUBLISHED: July 23, 1993 (19930723)
INVENTOR(s): SAEKI KENICHI
APPLICANT(s): KYUSHU NIPPON DENKI SOFTWARE KK [000000] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 03-359735 [JP 91359735]
FILED: December 29, 1991 (19911229)
INTL CLASS: [5] **G06F-009/06**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 1638, Vol. 17, No. 602, Pg. 78, November 05, 1993 (19931105)

ABSTRACT

PURPOSE: To provide a program patch correcting system which can easily and

speedily execute exact correction.

CONSTITUTION: A segment selecting means 2 selects the segment of a program to be corrected by a user, and a **code** part **updating instruction assemble** means 3 inputs an **instruction** or the like to be **updated** in an **assembly** language by the user and the instruction is connected into a code in a machine word. Next, a code part designating address correction possibility discriminating means 4 judges whether the code to be corrected can be overwritten at a correcting position on the program or not and when the code can be overwritten, an **updating instruction** writing means 6 writes the code to be corrected in the area of a changeable instruction sentence on the program. When the code can not be overwritten, a code part unused area retrieving means 8 secures an area for correcting the code part, and a code part update data writing means 9 writes the code to be corrected in the secured area.

17/5/13 (Item 13 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

03857236 **Image available**
PROGRAM OPTIMIZATION PROCESSOR

PUB. NO.: 04-252336 [JP 4252336 A]
PUBLISHED: September 08, 1992 (19920908)
INVENTOR(s): OZAWA TOSHIHIRO
APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 03-008835 [JP 918835]
FILED: January 29, 1991 (19910129)
INTL CLASS: [5] **G06F-009/38** ; **G06F-009/45**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 1473, Vol. 17, No. 33, Pg. 4, January 21, 1993 (19930121)

ABSTRACT

PURPOSE: To obtain a program optimization processor which changes the executing order of instruction trains so as to attain the optimization of a program to the execution of a processor which performs the pipeline processing in regard of the program optimization processing of a computer.
CONSTITUTION: An inter-instruction weight data base 2 holds the information which shows the estimated pipeline break length as the prescribed weight value when two instructions are continuously carried out for each arrangement of two necessary instructions of different types. A reliance analyzing part 3 shows the master-slave relation with the instruction right before the output of each input defined as a master instruction for each input of the corresponding instruction in regard of each instruction of an instruction train 1. Then the part 3 generates the information that set the weight decided by the base 2 as the master-slave relation information 5 for each master-slave relation. An instruction order setting part 4 repeats the prescribed instruction order setting processing until. the corresponding candidate **assembly** 6, becomes empty after the **instruction** lacking a master **instruction** is transferred to the candidate **assembly** from the relation 5 from the **assembly** 6. Thus a **new instruction** train is obtained.

17/5/14 (Item 14 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

03870533 **Image available**
PARALLEL OPTIMIZATION SYSTEM

PUB. NO.: 04-235633 [JP 4235633 A]
PUBLISHED: August 24, 1992 (19920824)
INVENTOR(s): NAKAMURA SAORI

APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 03-012390 [JP 9112390]
FILED: January 10, 1991 (19910110)
INTL CLASS: [5] **G06F-009/45**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 1463, Vol. 16, No. 586, Pg. 124, December 25, 1992 (19921225)

ABSTRACT

PURPOSE: To obtain an object program which excells in the executing speed.

CONSTITUTION: A definition/reference relation analyzing means 21 analyzes a fact whether an existing definition/reference **instruction assembly** and a **new** definition/reference **instruction assembly** are included in a parallel **assembly instruction** train 1 or not. An operand changing means 22 changes an operand so as to use an unused register in place of a symbol for **instruction** included in the **new** definition/reference **instruction assembly** . An **instruction** shift means 23 shifts the **instruction** included in the new definition/reference **instruction assembly** so as to attain the parallel execution of the **instructions** included in both existing and new definition/reference **instruction assemblies** . A transfer **instruction** adding means 24 adds 8 transfer instruction to the train 1, and an idle parallel instruction deleting means 25 deletes the idle parallel instruction produced with the shift carried out by an instruction of the means 23 and generates an optimized parallel **assembly instruction** train 3.

17/5/15 (Item 15 from file: 347)

03788733
ASSEMBLER PROCESSING SYSTEM

PUB. NO.: 04-153833 [JP 4153833 A]
PUBLISHED: May 27, 1992 (19920527)
INVENTOR(s): SHIMAZU RIEKO
ARAKI SATORU
MUROTA NORIE
APPLICANT(s): PFU LTD [366680] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 02-280321 [JP 90280321]
FILED: October 18, 1990 (19901018)
INTL CLASS: [5] **G06F-009/45**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 1421, Vol. 16, No. 442, Pg. 76, September 16, 1992 (19920916)

ABSTRACT

PURPOSE: To make programming and **assembler** processing efficient and fast by using signed numerals and unsigned numerals together according to whether or not signs are added to constant descriptions in an **assembler** language **program** .

CONSTITUTION: For the description format of constants, a signal which uses a specific character can be added to the head of unsigned numeric expression, an **assembler** discriminates between a signed numeral and an unsigned numeral according to whether or not the sign is **added** to a numeral, and **instructions** are converted into machine word instructions while classified into instructions for signed processing and instructions for unsigned processing. The signed and unsigned constants can be used together in a source program without increasing the processing load on the **assembler** much. Consequently, the programming and **assembler** processing can be made efficient and fast.

17/5/27 (Item 27 from file: 347)
DIALOG(R)File 347:JAPIO

01991941 **Image available**
EXTENSION METHOD FOR PROGRAM LANGUAGE

PUB. NO.: 61-206041 [JP 61206041 A]
PUBLISHED: September 12, 1986 (19860912)
INVENTOR(s): MARUYAMA SADANOBU
SUZUKI NAOYA
APPLICANT(s): SONY CORP [000218] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 60-047813 [JP 8547813]
FILED: March 11, 1985 (19850311)
INTL CLASS: [4] **G06F-009/44**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 543, Vol. 11, No. 37, Pg. 17, February 04, 1987 (19870204)

ABSTRACT

PURPOSE: To obtain an extension method for a program language which can add easily a **new instruction** and to use it immediately by **assembling** the file of a **new instruction** to a program language itself when the program language starts to operate.

CONSTITUTION: From a keyboard 2 to a **compiler** 1, a source **program** is given, **compiled** successively to an object **program** (ObjP), and the source program is successively stored to a memory 4A and the compiled ObjP is successively stored to a memory area 4B respectively. For the program execution including the **new instruction** , the program of a disk 5 is loaded and a name file 8 is retrieved. Thus, the file name of modules 9 and 10 to add the **new additional instruction** is known. For that reason, the corresponding module 9 is accessed, and **assembled** to the **program** language itself. In the same manner, the module 10 is accessed and **assembled** to the **program** language itself. Hereinafter, the retrieval of the name file 8 is continued, the file to add all the **additional instructions** is **assembled** to the **program** language itself and thereafter, execution is performed.

17/5/55 (Item 26 from file: 350)
DIALOG(R)File 350:Derwent WPIX

011434308 **Image available**
WPI Acc No: 1997-412215/199738
XRPX Acc No: N97-343409
C language program storing method to ROM for electronic control unit of vehicle - by transferring deleted data **content to** assembler instruction **sentence, that deletes** function **call in program loop without affecting control process, during formation of C language program**
Patent Assignee: FUJITSU TEN LTD (FUTE)
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 9185512	A	19970715	JP 95343762	A	19951228	199738	B

Priority Applications (No Type Date): JP 95343762 A 19951228
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 9185512	A		5		

Abstract (Basic): JP 9185512 A

The method involves adding an **assembler instruction** sentence, that has different absolute address instruction, relative address instruction, and data content, to a loop function call. The loop function call is used during the development of a program described in

a C language.

A control process is not affected by the **added assembler instruction** sentence which deletes the **function** call arranged in the program loop. The deleted **data** content is transferred to the **assembler instruction** sentence during the formation of the C language program.

ADVANTAGE - Fixes address of ROM freely when changing loop of C language **program** in **assembler** language sentence.

Dwg.1/7

Title Terms: LANGUAGE; PROGRAM; STORAGE; METHOD; ROM; ELECTRONIC; CONTROL; UNIT; VEHICLE; TRANSFER; DELETE; DATA; CONTENT; **ASSEMBLE** ; INSTRUCTION; SENTENCE; DELETE; FUNCTION; CALL; PROGRAM; LOOP; AFFECT; CONTROL; PROCESS ; FORMATION; LANGUAGE; PROGRAM

Derwent Class: T01; X22

International Patent Class (Main): **G06F-009/45**

International Patent Class (Additional): **G06F-009/06**

File Segment: EPI

17/5/56 (Item 27 from file: 350)

DIALOG(R)File 350:Derwent WPIX

011394477 **Image available**

WPI Acc No: 1997-372384/199734

XRPX Acc No: N97-309294

Method of dynamically allocating processing element instruction to processing element in SIMD processor - involves updating instruction **decoder configuration according to control specification if necessary for processing element**

Patent Assignee: MOTOROLA INC (MOTI)

Inventor: ACOSTA A C; GEHMAN J B; STEENSTRA M E

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5649179	A	19970715	US 95444637	A	19950519	199734	B

Priority Applications (No Type Date): US 95444637 A 19950519

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5649179	A		9		

Abstract (Basic): US 5649179 A

The method involves parsing a source code instruction for the SIMD processor into components applicable to the processing element **instruction** . The components are **assembled** into a control specification for the processing element. An **instruction** decoder configuration is **updated** according to the control specification if necessary for the processing element.

The processing element instruction is determined for the processing element an array representing the control specification is provided, such that elements of the array represent digital control signals for the processing element. All of the elements of the array are initialised to a symbol representing that values of the elements are undefined.

ADVANTAGE - Provides programming flexibility. Minimises SIMD processor instruction word length.

Dwg.5/5

Title Terms: METHOD; DYNAMIC; ALLOCATE; PROCESS; ELEMENT; INSTRUCTION; PROCESS; ELEMENT; SIMD; PROCESSOR; UPDATE; INSTRUCTION; DECODE; CONFIGURATION; ACCORD; CONTROL; SPECIFICATION; NECESSARY; PROCESS; ELEMENT

Derwent Class: T01

International Patent Class (Main): **G06F-015/00**

File Segment: EPI

17/5/57 (Item 28 from file: 350)

DIALOG(R)File 350:Derwent WPIX

011372697 **Image available**
WPI Acc No: 1997-350604/199732
XRPX Acc No: N97-290669
Program **optimisation device for** assembler program **- performs virtual command execution and determines when current instruction is same as** previous instruction **and deleting current** instruction **as unnecessary**
Patent Assignee: MATSUSHITA ELECTRIC IND CO LTD (MATU)
Inventor: HASIGUTI W
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5644769	A	19970701	US 94258990	A	19940613	199732	B
			US 96699220	A	19960819		

Priority Applications (No Type Date): JP 93141728 A 19930614
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5644769	A		11		Cont of application US 94258990

Abstract (Basic): US 5644769 A
The device includes an extraction device which extracts a value of an internal state of a computer by virtually executing an instruction of the program prior to actual execution of the program. The value of the internal state of the computer is a value of a register extracted by the virtual execution of an instruction. The extracted value is stored. A judgement device determines when an instruction is unnecessary by comparing the extracted stored value with at least a part of a value extracted and stored for a **prior instruction** .
When the value extracted from the virtual execution of the instruction is identical to the value extracted from the **prior instruction** , the **instruction** is deemed unnecessary. Unnecessary instructions are deleted. The value of the internal state of the computer is made indefinite when the extraction device fails to extract a value for an instruction to be executed.
ADVANTAGE - Optimises program operation by preventing overlapping of instructions.
Dwg.2/5
Title Terms: PROGRAM; OPTIMUM; DEVICE; **ASSEMBLE** ; PROGRAM; PERFORMANCE; VIRTUAL; COMMAND; EXECUTE; DETERMINE; CURRENT; INSTRUCTION; INSTRUCTION; DELETE; CURRENT; INSTRUCTION; UNNECESSARY
Derwent Class: T01
International Patent Class (Main): **G06F-009/44**
File Segment: EPI

17/5/58 (Item 29 from file: 350)
DIALOG(R)File 350:Derwent WPIX

011025664 **Image available**
WPI Acc No: 1997-003588/199701
XRPX Acc No: N97-003171
Automatic debugging command file formation for computer system - by re-assembling **corrected source** program **counter which includes implanting debugging** command **at** new **opposed-position in automatically generating** new command **file with** new **brake point** command
Patent Assignee: HITACHI LTD (HITA); HITACHI AMERICA LTD (HITA)
Inventor: SHRIDHAR A; SIMONS J
Number of Countries: 002 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 8272648	A	19961018	JP 95285382	A	19951101	199701	B
US 5815714	A	19980929	US 94366050	A	19941229	199846	
			US 97839229	A	19970421		

Priority Applications (No Type Date): US 94366050 A 19941229; US 97839229 A 19970421
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 8272648	A		13	G06F-011/28	
US 5815714	A			G06F-009/45	Cont of application US 94366050

Abstract (Basic): JP 8272648 A

The method involves embedding one debugging command to one line of a source program counter. A brake point command is formed corresp. to each extracted implanting debugging command. A command file is generated by writing the brake point command and the debugging command to a debugging command file. An object code is formed from the source program command which is retouched by removing the brake point command and the implanting debugging command from the debugging command file.

A **new command** file which includes the **new** brake point **command** corresp. to the implanting debugging command, is generated. The implanting debugging **command** is in a **new** opposed-position on the source **program** counter during re- **assembling** of the corrected source **program** counter.

ADVANTAGE - Enables automatic formation of debugging command file utilised by debugger during simulation execution of object code drawn from source program counter.

Dwg.2/8

Title Terms: AUTOMATIC; DEBUG; COMMAND; FILE; FORMATION; COMPUTER; SYSTEM; **ASSEMBLE** ; CORRECT; SOURCE; PROGRAM; COUNTER; IMPLANT; DEBUG; COMMAND; NEW; OPPOSED; POSITION; AUTOMATIC; GENERATE; NEW; COMMAND; FILE; NEW; BRAKE; POINT; COMMAND
Derwent Class: T01
International Patent Class (Main): **G06F-009/45** ; **G06F-011/28**
File Segment: EPI

17/5/59 (Item 30 from file: 350)
DIALOG(R)File 350:Derwent WPIX

010828605 **Image available**
WPI Acc No: 1996-325557/199633
XRPX Acc No: N96-274068

Instruction change and insertion method for object program counter - by changing object program to source program, forming meaning label corresp. to each reference address of instruction , assembling **source** program **after insertion or changing of instruction and converting source program to object program**

Patent Assignee: SUMITOMO METAL IND LTD (SUMQ)
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 8147155	A	19960607	JP 94288352	A	19941122	199633	B

Priority Applications (No Type Date): JP 94288352 A 19941122
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 8147155	A		7	G06F-009/06	

Abstract (Basic): JP 8147155 A

The method involves reverse **assemble** of an object **program** (11) and changing the object program to a source program (13). The instruction including the reference address which analyses a source program is detected. A meaning label is formed and replaced for each reference address. The meaning label are stored in a memory (3).

The instruction for the change or insertion is detected from the source program. When a **new instruction** is inserted before the **instruction** which gave the meaning label, a command acquisition stage (4) changes the head of the **instruction** which inserts the label. A **assembler** (5) **assembles** the source **program** which changes or

inserts the instruction and converts the source program to the object program.

ADVANTAGE - Eases changing or inserting of instruction. Raises flexibility of instruction change or insertion.

Dwg.1/5

Title Terms: INSTRUCTION; CHANGE; INSERT; METHOD; OBJECT; PROGRAM; COUNTER; CHANGE; OBJECT; PROGRAM; SOURCE; PROGRAM; FORMING; MEANING; LABEL; CORRESPOND; REFERENCE; ADDRESS; INSTRUCTION; **ASSEMBLE** ; SOURCE; PROGRAM; AFTER; INSERT; CHANGE; INSTRUCTION; CONVERT; SOURCE; PROGRAM; OBJECT; PROGRAM

Derwent Class: T01

International Patent Class (Main): **G06F-009/06**

File Segment: EPI

17/5/60 (Item 31 from file: 350)

DIALOG(R)File 350:Derwent WPIX

010137423 **Image available**

WPI Acc No: 1995-038674/199506

XRPX Acc No: N95-030620

Language processing system for code processing in high-level language for computer - has code optimisation unit and code generating unit for generating machine language code from optimised intermediate code

Patent Assignee: NEC CORP (NIDE)

Inventor: MIZUSE H

Number of Countries: 004 Number of Patents: 005

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 633526	A2	19950111	EP 94110522	A	19940706	199506	B
EP 633526	A3	19950607	EP 94110522	A	19940706	199610	
US 5832273	A	19981103	US 94267911	A	19940706	199851	
EP 633526	B1	20011205	EP 94110522	A	19940706	200203	
DE 69429305	E	20020117	DE 629305	A	19940706	200213	
			EP 94110522	A	19940706		

Priority Applications (No Type Date): JP 93167584 A 19930707

Cited Patents: No-SR.Pub; 4.Jnl.Ref

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 633526	A2	E	20	G06F-009/45	
Designated States (Regional): DE FR GB					
EP 633526	A3			G06F-009/45	
US 5832273	A			G06F-009/45	
EP 633526	B1	E		G06F-009/45	
Designated States (Regional): DE FR GB					
DE 69429305	E			G06F-009/45	Based on patent EP 633526

Abstract (Basic): EP 633526 A

The language processing system includes an instruction passing portion (103) for inputting a source program (101) written by a high level language and an **assembler** language. The system further includes an intermediate code generator (105), a code optimiser (108), a code generator (110), a register information discriminating portion (104) and a register information storage portion (107).

The code optimiser performs optimisation by deleting a register operating instruction w.r.t. the register operating instruction code in the intermediate code when the register value indicated in it is the same as that indicated by an immediately preceding register information or an immediately **preceding** register operating **instruction** code.

USE/ADVANTAGE - Code processing in high-level language as problem language for microcomputer. Permits redundant instruction code for reducing **code** length of object **program** when **assembler** source is inserted in high-level language program.

Dwg.1/11

Title Terms: LANGUAGE; PROCESS; SYSTEM; CODE; PROCESS; HIGH; LEVEL; LANGUAGE; COMPUTER; CODE; OPTIMUM; UNIT; CODE; GENERATE; UNIT; GENERATE;

MACHINE; LANGUAGE; CODE; OPTIMUM; INTERMEDIATE; CODE
Derwent Class: T01
International Patent Class (Main): **G06F-009/45**
File Segment: EPI

17/5/61 (Item 32 from file: 350)
DIALOG(R)File 350:Derwent WPIX

010047852 **Image available**
WPI Acc No: 1994-315563/199439
Related WPI Acc No: 1993-295490
XRPX Acc No: N94-247827

Symbolic ladder logic programming with automatic attachment of symbols to PLC addresses - receiving command from operator selecting symbol from displayed list of symbols, and modifying operator-identified instruction in program so that it references selected symbol

Patent Assignee: ICOM INC (ICOM-N)
Inventor: MENTER J J; ZIFFERER S C
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5349518	A	19940920	US 89374487	A	19890630	199439	B

Priority Applications (No Type Date): US 89374487 A 19890630
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5349518	A		43	G06F-015/00	

Abstract (Basic): US 5349518 A

The method involves maintaining a symbol assignment record in the computer, which identifies a correspondence between symbols and elements of the data tables. Each element is assigned to a specific symbol and each symbol represents a specific element, so that instructions in the ladder logic program reference an element using a symbol representing the element. A list of symbols obtained from the symbol assignment record are displayed on the monitor while the program is being edited. A command is received from the operator selecting a symbol from the displayed list of symbols, and an operator-identified instruction in the program is modified so that it references the selected symbol.

Commands are received from the operator specifying a portion of a desired symbol, and in the list of symbols only those symbols from the symbol assignment record which match the portion of the desired symbol are displayed. Commands are received from the **operator** identifying **additional** portions of the desired symbol, and in the list of symbols only those symbols from the symbol assignment record are which match the additional portions of the desired symbol are displayed.

USE/ADVANTAGE - For performing control **functions** for **assembly** line machines, machine tools, and other types of industrial equipment. Provides productivity aids for ladder logic programmer.

Dwg.27/37

Title Terms: SYMBOL; LADDER; LOGIC; PROGRAM; AUTOMATIC; ATTACH; SYMBOL; PLC ; ADDRESS; RECEIVE; COMMAND; OPERATE; SELECT; SYMBOL; DISPLAY; LIST; SYMBOL; MODIFIED; OPERATE; IDENTIFY; INSTRUCTION; PROGRAM; SO; REFERENCE; SELECT; SYMBOL
Index Terms/Additional Words: PROGRAMMABLE; LOGIC; CONTROLLER
Derwent Class: T01
International Patent Class (Main): **G06F-015/00**
File Segment: EPI

17/5/63 (Item 34 from file: 350)
DIALOG(R)File 350:Derwent WPIX

009805006 **Image available**

WPI Acc No: 1994-084861/199411
XRPX Acc No: N94-066435

Programmable controller with ladder diagram macro instructions - uses ladder logic processor for high speed execution of common ladder logic instructions and microprocessor for execution of remaining logic instructions

Patent Assignee: ALLEN BRADLEY CO (ALLB)
Inventor: BROOKS J W; KOLAT J J; YOKE M D
Number of Countries: 004 Number of Patents: 006
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 586813	A2	19940316	EP 93110447	A	19930630	199411	B
US 5295059	A	19940315	US 92942254	A	19920909	199411	
EP 586813	A3	19960508	EP 93110447	A	19930630	199628	
EP 586813	B1	19980805	EP 93110447	A	19930630	199835	
DE 69320122	E	19980910	DE 620122	A	19930630	199842	
			EP 93110447	A	19930630		
EP 586813	B2	20020612	EP 93110447	A	19930630	200239	

Priority Applications (No Type Date): US 92942254 A 19920909
Cited Patents: No-SR.Pub; 1.Jnl.Ref; EP 28068; EP 312611; EP 503256; JP 62020029; US 4302820
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 586813	A2	E	13	G05B-019/05	
Designated States (Regional): DE GB IT					
US 5295059	A		11	G06F-015/46	
EP 586813	A3			G05B-019/05	
EP 586813	B1	E		G05B-019/05	
Designated States (Regional): DE GB IT					
DE 69320122	E			G05B-019/05	Based on patent EP 586813
EP 586813	B2	E		G05B-019/05	
Designated States (Regional): DE GB IT					

Abstract (Basic): EP 586813 A

The programmable controller comprises a first section of memory for storing a ladder logic control program. The ladder logic program has a number of ladder logic instructions. One such logic instruction is a macro which specifies an operation code. A ladder logic processor executes a subset of the number of ladder logic instructions in which a subset includes the macro instruction.

The microprocessor executes the ladder logic instructions which the ladder logic processor in unable to execute. The first and second files and the result produced by executing the macro instruction are stored in a second section of memory. A third section of memory stores a macro instruction routine, having a number of ladder logic instructions which are executed by the ladder logic processor. The ladder logic processor, microprocessor and first, second and third sections of memory are connected together to enable the exchange of data and program instructions.

USE/ADVANTAGE - For complex manufacturing processes for controlling e.g **assembly** line or machine tool. Custom **instructions** can be **added** to set of programming **instructions** that are executable by program controller.

Dwg.1/7

Title Terms: PROGRAM; CONTROL; LADDER; DIAGRAM; MACRO; INSTRUCTION; LADDER; LOGIC; PROCESSOR; HIGH; SPEED; EXECUTE; COMMON; LADDER; LOGIC; INSTRUCTION; MICROPROCESSOR; EXECUTE; REMAINING; LOGIC; INSTRUCTION
Derwent Class: T01; T06; X25
International Patent Class (Main): G05B-019/05; **G06F-015/46**
International Patent Class (Additional): G05B-019/00
File Segment: EPI

17/5/64 **(Item 35 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

009409618 **Image available**
WPI Acc No: 1993-103129/199313
XRPX Acc No: N93-078403
Computer accessory to transform interactive terminals such as Minitel into micro-computers - uses unit additional to terminal with microprocessor and memories holding software common to many terminal types for translation into particular assembler **required**
Patent Assignee: INFOTEL MARKETING SARL (INFO-N)
Inventor: BATTIST L; DECH C; GARNIER G; IBOT A
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
FR 2679674	A1	19930129	FR 919384	A	19910724	199313	B

Priority Applications (No Type Date): FR 919384 A 19910724
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
FR 2679674	A1		20	G06F-013/10	

Abstract (Basic): FR 2679674 A

The terminal (32) has a central communications unit (62) with screen (64), keyboard (66), modem (68) and data interface (70). The accessory device (30) comprises a micro-processor (36) and its operating system, a read only memory (38) holding application software (52) and an input/output unit (42). It also contains random access memory (40) holding a second application program. Application software consist of a series of intermediate codes common to a number of different models of terminal. It is also independent of the **assembler** language of the micro-processor. The ROM also contains conversion software (56, 58, 60) consisting of two sub- **assemblies** . These are an interpretation stage (56) to translate the intermediate codes and an operating stage (58, 60) to transform the **primary instructions** into **assembler** language sequences directly executable by the micro-processor.

ADVANTAGE - Provides once-and-for-all development of application software used without modification on number of terminal models.

Dwg.2/2

Title Terms: COMPUTER; ACCESSORY; TRANSFORM; INTERACT; TERMINAL; MICRO; COMPUTER; UNIT; ADD; TERMINAL; MICROPROCESSOR; MEMORY; HOLD; SOFTWARE; COMMON; TERMINAL; TYPE; TRANSLATION; **ASSEMBLE** ; REQUIRE
Derwent Class: T01
International Patent Class (Main): **G06F-013/10**
International Patent Class (Additional): **G06F-013/12**
File Segment: EPI

17/5/65 (Item 36 from file: 350)
DIALOG(R)File 350:Derwent WPIX

009294930
WPI Acc No: 1992-422340/199251
XRPX Acc No: N92-322153
Compiler **for** program **patches created from source program - receives two input source files, one being source of program targetted by patch, other being source of patch and control instructions**
Patent Assignee: ANONYMOUS (ANON)
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
RD 343068	A	19921110	RD 92343068	A	19921020	199251	B

Priority Applications (No Type Date): RD 92343068 A 19921020
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
RD 343068	A		1	G06F-000/00	

Abstract (Basic): RD 343068 A

A program patch is a set of instructions inserted into an existing object program by a process of inspecting and modifying data called ''zap''. Commonly, existing instructions are modified to branch to a ''patch area'' (an idle section of the object), the new code is inserted into the patch area, the **original** modified **instructions** are recoded (if needed) at the patch area, etc.. Preparation of a patch is a manual, error-prone process. This fact has historically affected the quality and limited the size and applicability of program patches.

The patch compiler has all the capabilities of an ordinary compiler. It receives two input source files, rather than one. One of the files is the source of the program which will be the target of the patch. The other file contains the source for the patch instructions, as well as control **instructions** . The **compiler** re-processes the original source, recreates necessary symbol tables (including all existing offsets), and then compiles the patch with reference to the named objects in the original source. The output of the process includes, in addition to ordinary object code, control statements for the ''zap'' program.

USE/ADVANTAGE - Can use a listing (printout file) of the program version which is to be modified by patch. Provides higher level of confidence that generated patch hits correct target. For use with **Assembler** programs as well as programs in a High Level Language.

Dwg.0/0

Title Terms: COMPILE; PROGRAM; PATCH; SOURCE; PROGRAM; RECEIVE; TWO; INPUT; SOURCE; FILE; ONE; SOURCE; PROGRAM; TARGET; PATCH; SOURCE; PATCH; CONTROL ; INSTRUCTION

Derwent Class: T01

International Patent Class (Main): **G06F-000/00**

File Segment: EPI

17/5/69 (Item 40 from file: 350)

DIALOG(R)File 350:Derwent WPIX

004357005

WPI Acc No: 1985-183883/198530

XRPX Acc No: N85-138066

Data processing system with memory hierarchy - has ROM for storing program to be executed, changeable memory for storing instructions to change program, and RAM for revised program

Patent Assignee: NCR CORP (NATC)

Inventor: HILBRINK J O

Number of Countries: 006 Number of Patents: 007

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 8503150	A	19850718	WO 85US2096	A	19850718	198530	B
EP 167572	A	19860115	EP 85900439	A	19850111	198603	
JP 61500991	W	19860515	JP 85500216	A	19860814	198626	
CA 1224573	A	19870721				198733	
US 4769767	A	19880906	US 84567486	A	19840103	198838	
EP 167572	B	19890823	EP 84900439	A	19841224	198934	
DE 3479539	G	19890928				198940	

Priority Applications (No Type Date): US 84567486 A 19840103

Cited Patents: 2.Jnl.Ref; DE 2400244; DE 3210616; EP 48816; FR 2427646; JP 58107931; EP 38816

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 8503150	A	E	36		
	Designated States (National): JP				
	Designated States (Regional): DE FR GB				
EP 167572	A	E			
	Designated States (Regional): DE FR GB				
EP 167572	B	E			
	Designated States (Regional): DE FR GB				

Abstract (Basic): WO 8503150 A

The system consists of a central processing unit and three sets of semiconductor memories. The first memory consists of a very large scale non programmable read only memory and is used to store the computer programme. The second memory is an erasable programmable read only memory and is used to store instructions to change the computer programme. The third memory is a read-write type memory.

The **original** computer programme and the modifying **instructions** are loaded, from their respective memories, into the third memory which now contains the revised computer programme to be executed by the central processing unit.

ADVANTAGE - Revisions to compute programms can be made faster. Cheaper storage method is used. Program available after power failure.

0/6

Title Terms: DATA; PROCESS; SYSTEM; MEMORY; HIERARCHY; ROM; STORAGE; PROGRAM; EXECUTE; CHANGE; MEMORY; STORAGE; INSTRUCTION; CHANGE; PROGRAM; RAM; REVISED; PROGRAM

Derwent Class: T01

International Patent Class (Additional): **G06F-007/00** ; **G06F-009/44** ; **G06F-012/06** ; **G06F-013/00** ; G11C-011/34

File Segment: EPI

18/5/1 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO

05667378 **Image available**
ASSEMBLER CONVERTER

PUB. NO.: 09-282178 [JP 9282178 A]
PUBLISHED: October 31, 1997 (19971031)
INVENTOR(s): KAWADA SHINYA
APPLICANT(s): FUJI ELECTRIC CO LTD [000523] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 08-111974 [JP 96111974]
FILED: April 09, 1996 (19960409)
INTL CLASS: [6] **G06F-009/45**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JAPIO KEYWORD: R131 (INFORMATION PROCESSING -- Microcomputers & Microprocessers)

ABSTRACT

PROBLEM TO BE SOLVED: To improve the efficiency of optimized assembling by tentatively converting a source program to a source intermediate language, allocating a memory corresponding to the attribute and alternately rearranging the memory and an address so as to minimize the number of times of memory access and making access time shortest.

SOLUTION: High-level program language algorithm such as the numerical expression of an assembling object or FORTRAN or the like is converted to the source intermediate language first and a file for specifying the attribute of the data is prepared (step 10). Then, the allocation of memory addresses is performed to four levels, the addresses are replaced and an intermediate language is optimized (step 11). The optimized intermediate language is converted to a mnemonic language and general purpose **cross assembler** conversion is performed.

18/5/2 (Item 2 from file: 347)
DIALOG(R)File 347:JAPIO

02907742 **Image available**
DEVELOPMENT SYSTEM FOR GENERAL PURPOSE CROSS SOFTWARE

PUB. NO.: 01-205342 [JP 1205342 A]
PUBLISHED: August 17, 1989 (19890817)
INVENTOR(s): KUNIMINE YUKIO
APPLICANT(s): SOFUTO UEA ASHISUTO KK [000000] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 63-028960 [JP 8828960]
FILED: February 12, 1988 (19880212)
INTL CLASS: [4] **G06F-009/44** ; **G06F-009/06** ; **G06F-011/28**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JOURNAL: Section: P, Section No. 960, Vol. 13, No. 508, Pg. 49, November 15, 1989 (19891115)

ABSTRACT

PURPOSE: To attain cross in respective MPUs and to permit the MPUs to attain speedy correspondence by compiling a source program in a general purpose cross C compiler, outputting an object program and debugging it in an ICE so as to make it into ROM.
CONSTITUTION: The source program for a C language, which is described manually by a user, is compiled in a general purpose cross C compiler 2, and an assembly source program is outputted. The program is assembled in a general **cross assembler** 1, and the objective object program is outputted. The program generated on a host computer makes down-load later/by the ICE and the like via the transmission means of an on line and the like. The program which has been debugged and outputted from the ICE is made into ROM, and it is integrated into a final product. Thus, software

for MPU is developed by cross software.

18/5/3 **(Item 3 from file: 347)**
DIALOG(R)File 347:JAPIO

00981947 **Image available**
DEVELOPMENT SUPPORTING DEVICE FOR COMPUTER PROGRAM

PUB. NO.: 57-132247 [JP 57132247 A]
PUBLISHED: August 16, 1982 (19820816)
INVENTOR(s): TANEMURA RYOHEI
APPLICANT(s): KOA DIGITAL KK [000000] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 56-017736 [JP 8117736]
FILED: February 08, 1981 (19810208)
INTL CLASS: [3] **G06F-009/06 ; G06F-003/16 ; G06F-009/00 ; G06F-011/00**

JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units); 45.3 (INFORMATION PROCESSING -- Input Output Units)
JAPIO KEYWORD: R108 (INFORMATION PROCESSING -- Speech Recognition & Synthesis); R131 (INFORMATION PROCESSING -- Microcomputers & Microprocessers)
JOURNAL: Section: P, Section No. 156, Vol. 06, No. 233, Pg. 23, November 19, 1982 (19821119)

ABSTRACT

PURPOSE: To support the development of a microcomputer program rapidly and accurately by automatically performing assembling, the operation of a simulator, etc., systematically and consistently.

CONSTITUTION: A **cross assembler** permits a source program to be read in through a paper tape reader PTR5 or from a floppy disk 12, thereby outputting an object program in prescribed format to a PTR6 or the disk 12. A simulator, on the other hand, permits the object program to be read through the PTR5 or from the disk 12 to simulate the conditions of various registers of a microcomputer, an RAM, an I/O port, etc., by instruction codes, and the results are displayed on a CRT7, thereby tracing the program. Those respective operations are performed systematically and consistently including the inspecting operation of each circumferential equipment.

18/5/5 **(Item 2 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

011606773 **Image available**
WPI Acc No: 1998-023901/199803
XRPX Acc No: N98-018443
Assembler converter for expression or high class language program conversion in CPU chip - uses cross assemble **program to convert mnemonic language, which is rewritten from intermediate language after end of CPU memory and address allocation and recombination, into computer program**
Patent Assignee: FUJI ELECTRIC CO LTD (FJIE)
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 9282178	A	19971031	JP 96111974	A	19960409	199803	B

Priority Applications (No Type Date): JP 96111974 A 19960409
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 9282178	A		5	G06F-009/45	

Abstract (Basic): JP 9282178 A

The converter has a designating unit that designates data for conversion into input and output attributes, variable or temporary variables to produce an attribute file. Address allocating units of different levels allocate the memory address of the chip memory and external memory of a CPU chip based on the intermediate language and attribute designation file.

An access controller reduces the access frequency of a memory of the CPU chip and rearranges the memory and its address to ensure quick access time. A rewriting unit rewrites the intermediate language in a mnemonic language after the allocation and recombination of the memory and its address has ended. A **cross assemble** program is used to convert the rewritten mnemonic language into a compute program.

ADVANTAGE - Improves assembly efficiency due to optimisation done by minimising memory access frequency and rearranging memory and its address.

Dwg.3/5

Title Terms: ASSEMBLE; CONVERTER; EXPRESS; HIGH; CLASS; LANGUAGE; PROGRAM; CONVERT; CPU; CHIP; CROSS; ASSEMBLE; PROGRAM; CONVERT; MNEMONIC; LANGUAGE ; REWRITING; INTERMEDIATE; LANGUAGE; AFTER; END; CPU; MEMORY; ADDRESS; ALLOCATE; RECOMBINATION; COMPUTER; PROGRAM

Index Terms/Additional Words: CENTRAL; PROCESSING; UNIT

Derwent Class: T01

International Patent Class (Main): **G06F-009/45**

File Segment: EPI

18/5/6 (Item 3 from file: 350)

DIALOG(R)File 350:Derwent WPIX

004716566

WPI Acc No: 1986-219908/198634

XRPX Acc No: N86-164146

Emulator development processing system - includes various processors selected as headed to avoid head for cross - assemblers

Patent Assignee: TOSHIBA KK (TOKE)

Inventor: NAMIMOTO K

Number of Countries: 004 Number of Patents: 004

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 191402	A	19860820	EP 86101425	A	19860204	198634	B
EP 191402	B	19910724				199130	
DE 3680349	G	19910829				199136	
US 5101342	A	19920331	US 89353175	A	19890516	199216	

Priority Applications (No Type Date): JP 8521069 A 19850206

Cited Patents: 4.Jnl.Ref; A3...8830; EP 135753; No-SR.Pub

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 191402	A	E	18		
	Designated States (Regional): DE FR GB				
EP 191402	B				
	Designated States (Regional): DE FR GB				
US 5101342	A		6		

Abstract (Basic): EP 191402 B

The system has different processing unit each receiving a rest mode signal and each made inoperative in response to the rest mode signal. A bus device has address, data and control buses. A memory stores execution programs for the processing units. A control selectively actuates the processing units. The selection control has a section with a specified address receiving and storing selection data and supplies the rest mode signal to all input terminals of the processing unit except the one designated by the stored selection data.

The selection data is generated from the active processing unit. The memory is connected through the bus.

ADVANTAGE - Structures of control hardware and software to activate selected processor are simple. (18pp Dwg.No.0/1)

Title Terms: EMULATION; DEVELOP; PROCESS; SYSTEM; VARIOUS; PROCESSOR; SELECT; HEAD; AVOID; HEAD; CROSS; ASSEMBLE
Derwent Class: T01
International Patent Class (Additional): **G06F-009/44** ; G06G-001/24; G06G-009/45
File Segment: EPI

18/5/7 (Item 4 from file: 350)
DIALOG(R)File 350:Derwent WPIX

003084407
WPI Acc No: 1981-J4449D/198136
Data processing device using subroutine call instruction - has address field coded and applied to data converter with address field connected to instruction register
Patent Assignee: TOKYO SHIBAURA DENKI KK (TOKE)
Inventor: KITAGAWA Y; MORIYA Y
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 4285036	A	19810818				198136 B

Priority Applications (No Type Date): JP 7810930 A 19780202
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 4285036	A		9		

Abstract (Basic): US 4285036 A

An ordinary subroutine CALL instruction is constructed by four words with one word comprising 4 bits. However, during assembly by the **cross - assembler** , if the subroutine CALL instruction to be constructed by two words is defined by using a psuedo instruction, it may be converted into an object code instruction of two words. In the execution of the subroutine CALL instruction of two words with coded entry points, the effective address is restored by using a data converter having previously stored an input code and the effective address corresponding to it.

To achieve the above object, the data processing device comprises an ROM for storing instructions or data; an instruction register connected to the ROM for registering an instruction outputted from the ROM; an instruction decoder connected to the instruction register which decodes an instruction registered in the instruction register to produce various control signals; data converter connected to the instruction register which decodes a subroutine call instruction of one byte registered in the instruction register with the address field coded to produce a given program execution address; a program counter connected to the data converter for storing the program execution address outputted from the data converter; and a stack connected to the program counter for saving a return address of the subroutine.

2

Title Terms: DATA; PROCESS; DEVICE; SUBROUTINES; CALL; INSTRUCTION; ADDRESS; FIELD; CODE; APPLY; DATA; CONVERTER; ADDRESS; FIELD; CONNECT; INSTRUCTION; REGISTER
Derwent Class: T01
International Patent Class (Additional): **G06F-009/40**
File Segment: EPI

```
File   8:Ei Compendex(R) 1970-2004/Jan W3
         (c) 2004 Elsevier Eng.  Info. Inc.
File  35:Dissertation Abs Online 1861-2004/Dec
         (c) 2004 ProQuest Info&Learning
File 202:Info. Sci. & Tech. Abs. 1966-2004/Jan 20
         (c) 2004 EBSCO Publishing
File  65:Inside Conferences 1993-2004/Jan W4
         (c) 2004 BLDSC all rts. reserv.
File   2:INSPEC 1969-2004/Jan W3
         (c) 2004 Institution of Electrical Engineers
File 233:Internet & Personal Comp. Abs. 1981-2003/Sep
         (c) 2003 EBSCO Pub.
File  94:JICST-EPlus 1985-2004/Jan W3
         (c)2004 Japan Science and Tech Corp(JST)
File 603:Newspaper Abstracts 1984-1988
         (c)2001 ProQuest Info&Learning
File 483:Newspaper Abs Daily 1986-2004/Jan 28
         (c) 2004 ProQuest Info&Learning
File   6:NTIS 1964-2004/Jan W4
         (c) 2004 NTIS, Intl Cpyrght All Rights Res
File 144:Pascal 1973-2004/Jan W3
         (c) 2004 INIST/CNRS
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 1998 Inst for Sci Info
File  34:SciSearch(R) Cited Ref Sci 1990-2004/Jan W4
         (c) 2004 Inst for Sci Info
File  99:Wilson Appl. Sci & Tech Abs 1983-2004/Dec
         (c) 2004 The HW Wilson Co.
File 583:Gale Group Globalbase(TM) 1986-2002/Dec 13
         (c) 2002 The Gale Group
File 266:FEDRIP 2004/Dec
         Comp & dist by NTIS, Intl Copyright All Rights Res
File  95:TEME-Technology & Management 1989-2004/Jan W2
         (c) 2004 FIZ TECHNIK
File 438:Library Lit. & Info. Science 1984-2004/Dec
         (c) 2004 The HW Wilson Co

Set     Items   Description
S1     617995   ASSEMBL???
S2     103111   (ASSEMBL? OR COMPIL???)(5N)(CODE OR INSTRUCTION? ? OR PROG-
              RAM OR FILE OR DATA OR INFORMATION OR OPERATION? ? OR OPERATO-
              R? ? OR COMMAND? ? OR FUNCTION? ? OR DIRECTIVE? ? OR PROCEDUR-
              E? ?)
S3      24605   INSTRUCTION()SET()ARCHITECTURE? ? OR ISA
S4      37974   (NEW??? OR UPDAT? OR UPGRAD? OR RECENT? OR LATEST OR ADDED
              OR ADDITIONAL OR SUPPLEMENTARY OR EXTRA OR IMPROVED)(5N)(INST-
              RUCTION? ? OR OPERATOR? ? OR OPERAND? ? OR COMMAND? ?)
S5       6988   (OLD?? OR PREVIOUS? OR PRIOR OR PRECEDING OR FORMER? OR OR-
              IGINAL OR INITIAL OR PRIMARY)(5W)(INSTRUCTION? ? OR OPERATOR?
              ? OR OPERAND? ? OR COMMAND? ?)
S6     242730   (TRANSFORM? OR TRANSLAT? OR CONVERT??? OR CONVERSION? ? OR
              CHANG? OR MODIF???? OR MODIFICATION? ? OR AMEND? OR ADJUST???
              OR ADJUSTMENT? ? OR ALTER??? OR ALTERATION? ?)(5N)(INSTRUCTIO-
              N? ? OR OPERATOR? ? OR COMMAND? ? OR CODE OR DATA)
S7        240   S6(5N)(ASCII OR OPCODE OR OP()CODE)
S8        515   CROSS()ASSEMBL??? OR CROSSASSEMBL???
S9         11   S1 AND S7
S10        11   RD (unique items)
S11         2   S1 AND S2 AND S3 AND S4:S5
S12     19362   INSTRUCTION()SET? ?
S13        55   S1 AND S12 AND S4:S5
S14        55   S11 OR S13
S15        45   RD (unique items)
S16        39   S15 NOT PY=2001:2004
S17         4   S3:S5 AND S8
S18         3   RD (unique items)
S19      3000   AU=(SIMONS J? OR SIMONS, J?)
S20         0   S19 AND S2 AND S12
```

10/5/4 (Item 3 from file: 2)
DIALOG(R)File 2:INSPEC

1699292 INSPEC Abstract Number: C81021114
Title: **Direct-** assembler **with symbolic addressing**
Author(s): Nessler, N.
Journal: Funkschau no.5 p.87-92
Publication Date: 6 March 1981 Country of Publication: West Germany
CODEN: FUSHA2 ISSN: 0016-2841
Language: German Document Type: Journal Paper (JP)
Treatment: Theoretical (T)
Abstract: Describes the design and application of an **assembler** program which needs minimum storage space, does not require an intermediate memory buffer but allows symbolic addressing of jump and branch instructions. The **assembler** was generated for a 6502 microprocessor with a Baudot **code** input terminal but facilities for **conversion** to an **ASCII** terminal are provided. After initialization the program prompts the operator to enter the source code, checks the validity of the OP-code and addressing mode for the command (absolute, immediate, indirect, indexed etc.). The technique used for jump and branch to a label is explained, a list of special functions is provided, error messages are listed and the hexadecimal dump of the 1.5K byte **assembler** program, stored in a PROM is printed out. (0 Refs)
Subfile: C
Descriptors: microcomputers; program **assemblers**
Identifiers: **assembler** program; symbolic addressing; branch instructions; Baudot code input terminal; ASCII terminal; error messages; jump instructions
Class Codes: C5250 (Microcomputer techniques); C6150C (Compilers, interpreters and other processors)

10/5/6 (Item 2 from file: 233)
DIALOG(R)File 233:Internet & Personal Comp. Abs.

00190206 89PX04-006
Arguing with your computer Part 4 of a tutorial on machine code programming
Woods, Thomas B
PCM , April 1, 1989 , v6 n10 p58-64, 5 Pages
ISSN: 0747-0460
Languages: English
Document Type: Program Listing
assembly language program
Geographic Location: United States
Discusses binary-to- **ASCII conversion** . Presents **instructions** and an **assembler** program listing, ADD.ASM, to add two numbers supplied from the keyboard. Discusses passing arguments to executable programs, parsing lines, converting strings to binary, mnemonics, and subroutines. Article works in conjunction with previous articles to become a full tutorial on machine code programming. (bc)
Descriptors: Machine Language; Tutorial; Programming Instruction; Binary; ASCII
Identifiers: ADD.ASM

10/5/7 (Item 3 from file: 233)
DIALOG(R)File 233:Internet & Personal Comp. Abs.

00162479 88PI02-075
From ASCII strings to binary bits
Duncan, Ray
PC Magazine , Feb 16 1988 , v7 n3 p347-353, 6 Pages
ISSN: 0808-8507
Languages: English

Document Type: Column
assembly language program; C program
Geographic Location: United States
POWER PROGRAMMING column discusses C program routines that convert numeric ASCII strings into equivalent binary values. Presents programming instruction and **assembly** language program listings that perform the same conversions. (tjm)
Descriptors: **CONVERSIONS** ; **ASCII** ; PROGRAM LISTING; PROGRAMMING **INSTRUCTION**

10/5/10 (Item 1 from file: 99)
DIALOG(R)File 99:Wilson Appl. Sci & Tech Abs

1656199 H.W. WILSON RECORD NUMBER: BAST96021351
Converting binary to ASCII
Eisen, Alexander;
Electronic Design v. 44 (Mar. 18 '96) p. 144+
DOCUMENT TYPE: Feature Article ISSN: 0013-4872 LANGUAGE: English
RECORD STATUS: Corrected or revised record

ABSTRACT: Part of a special supplement on engineering software for software design engineers. An algorithm that **converts** binary to **ASCII code** in a loop of only a few lines of code is presented. The algorithm first converts the binary code to BCD, in which the decimal digitals are represented by nibbles, or half bytes. These BCD digits are then **converted** into **ASCII code** .

DESCRIPTORS: ASCII standard; Binary coded decimal system; **Assembly** language (Computer language);

10/5/11 (Item 2 from file: 99)
DIALOG(R)File 99:Wilson Appl. Sci & Tech Abs

1585173 H.W. WILSON RECORD NUMBER: BAST96056636
A closer look at instruction set design
Sibigtroth, James M;
Electronic Design v. 44 (Aug. 19 '96) p. 129-30+
DOCUMENT TYPE: Feature Article ISSN: 0013-4872 LANGUAGE: English
RECORD STATUS: Corrected or revised record

ABSTRACT: Part of a special supplement on engineering software for software design engineers. The writer addresses the question of what makes a good instruction set by examining some of the challenges faced by the designers of the instruction set for Motorola's new M68HC12 family of microcontrollers. The original motivation for the design of this instruction set was to support users of the M68HC11 that are pushing its upper performance limits in their highest-performance products. The new set had to accept all M68HC11 source code, but it had to be faster, so it was not expected to be the same cycle by cycle. Most of the **instructions** were the same, but some **changes** were made to the **opcode** map to enhance the indexed addressing mode. Even so, the same object code was produced, with the new instructions executed in fewer CPU cycles. The new CPU also had to support the modern realities of C programming and larger memory spaces. Fortunately, it was found that what is good for C is also good for **assembly** -language programmers, so it was not necessary to trade off between the 2 languages.

16/5/3 (Item 3 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

04189331 E.I. No: EIP95062747928
Title: Synthesis of application specific instruction sets
Author: Huang, Ing-Jer; Despain, Alvin M.
Corporate Source: Natl Sun Yat-Sen Univ, Kaohsiung, Taiwan
Source: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems v 14 n 6 Jun 1995. p 663-675
Publication Year: 1995
CODEN: ITCSDI ISSN: 0278-0070
Language: English
Document Type: JA; (Journal Article) Treatment: T; (Theoretical)
Journal Announcement: 9508W3
Abstract: An **instruction set** serves as the interface between hardware and software in a computer system. In an application specific environment, the system performance can be **improved** by designing an **instruction set** that matches the characteristics of hardware and the application. We present a systematic approach to generate application-specific **instruction sets** so that software applications can be efficiently mapped to a given pipelined micro-architecture. The approach synthesizes **instruction sets** from application benchmarks, given a machine model, an objective function, and a set of design constraints. In addition, **assembly** code is generated to show how the benchmarks can be compiled with the synthesized **instruction set** . The problem of designing **instruction sets** is formulated as a modified scheduling problem. A binary tuple is proposed to model the semantics of instructions and integrate the instruction formation process into the scheduling process. A simulated annealing scheme is used to solve for the schedules. Experiments have shown that the approach is capable of synthesizing powerful instructions for modern pipelined microprocessors, and running with reasonable time and a modest amount of memory for large applications. (Author abstract) 21 Refs.
Descriptors: *Computer architecture; Performance; Computer hardware; Computer software; Interfaces (computer); Pipeline processing systems; Standards; Functions; Constraint theory; Scheduling
Identifiers: Application specific **instruction set** ; Benchmarks; System performance; Machine model; Objection model; **Assembly** code; Binary tuple; Semantics; Pipelined microarchitecture
Classification Codes:
723.1 (Computer Programming); 722.2 (Computer Peripheral Equipment); 722.4 (Digital Computers & Systems); 902.2 (Codes & Standards); 921.6 (Numerical Methods)
722 (Computer Hardware); 723 (Computer Software); 902 (Engineering Graphics & Standards); 921 (Applied Mathematics)
72 (COMPUTERS & DATA PROCESSING); 90 (GENERAL ENGINEERING); 92 (ENGINEERING MATHEMATICS)

16/5/4 (Item 4 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

04159271 E.I. No: EIP95012516599
Title: POWER2 architecture and performance
Author: Hannon, E.L.; O'Connell, F.P.; Shieh, L.J.
Corporate Source: IBM Corp, Austin, TX, USA
Conference Title: Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors
Conference Location: Cambridge, MA, USA Conference Date: [illegible]-19941012
Sponsor: IEEE
E.I. Conference No.: 42124
Source: Proceedings - IEEE International Conference on Computer Design: VLSI in Computers and Processors 1994. IEEE, Piscataway, NJ, USA,94CH35712. p 336-339
Publication Year: 1994
CODEN: PIIPE6

Language: English
Document Type: CA; (Conference Article) Treatment: G; (General Review); T; (Theoretical)
Journal Announcement: 9507W2
Abstract: We briefly describe key features of the POWER2 architecture and the xlf compiler. Using FORTRAN loops and compiler generated instructions, we show how the compiler schedules **instructions** to make full use of **new** features of the POWER2 architecture. These loops are two of the most widely used Scientific/Engineering constructs. They demonstrate the ease with which one can attain near-peak or peak performance on the POWER2 architecture. Finally, we demonstrate the effect of these performance enhancements on several real applications. (Author abstract) 2 Refs.
Descriptors: Reduced **instruction set** computing; Computer systems; Performance; Program compilers; FORTRAN (programming language); Program **assemblers** ; Data structures; Computer circuits; Scheduling; Computer architecture
Identifiers: POWER2 architecture; XLF compiler
Classification Codes:
723.1.1 (Computer Programming Languages)
722.4 (Digital Computers & Systems); 723.2 (Data Processing); 723.1 (Computer Programming); 721.3 (Computer Circuits)
722 (Computer Hardware); 723 (Computer Software); 721 (Computer Circuits & Logic Elements)
72 (COMPUTERS & DATA PROCESSING)

16/5/5 (Item 5 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

04063951 E.I. No: EIP95022562304
Title: Risc microprocessors define computing's cutting edge
Author: Rubenstein, Roy H.
Source: New Electronics v 27 n 11 Nov 8 1994. p 58-60
Publication Year: 1994
CODEN: NWELAC ISSN: 0047-9624
Language: English
Document Type: JA; (Journal Article) Treatment: G; (General Review)
Journal Announcement: 9504W3
Abstract: A 64bit risc microprocessor can process large, complex applications and lessen the impact on performance of inevitable cache misses. To achieve outstanding performance, risc architectures make use of pipelining, breaking instructions into stages such that multiple instructions are overlapped, like an **assembly** line, each at a different stage of completion. Although the processing of each instruction requires several cycles, pipelining results in **instruction** execution occuring every cycle. The **latest** processors, while maintaining the pipeline concept, adopt more sophisticated superscalar designs - not only overlapping instruction sequentially, but also issuing multiple instruction in parallel. Future devices will obviously increase device's cache size and external bandwidth interface performance. 2 Refs.
Descriptors: Microcomputers; Reduced **instruction set** computing; Pipeline processing systems; Buffer storage; Performance; Multiprogramming; Large scale systems; Systems analysis; Computer software; Parallel processing systems
Identifiers: Internal architecture; Bus interface unit; Superscalar; Computing
Classification Codes:
722.4 (Digital Computers & Systems); 722.1 (Data Storage, Equipment & Techniques); 723.1 (Computer Programming); 714.2 (Semiconductor Devices & Integrated Circuits)
722 (Computer Hardware); 723 (Computer Software); 714 (Electronic Components)
72 (COMPUTERS & DATA PROCESSING); 71 (ELECTRONICS & COMMUNICATIONS)

16/5/6 (Item 6 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

03589055 E.I. Monthly No: EIM9304-019824
Title: Providing a laboratory for instruction set **design.**
Author: Nerheim-Wolfe, Rosalee
Corporate Source: DePaul Univ, Chicago, IL, USA
Conference Title: 23rd SIGCSE Technical Symposium on Computer Science Education
Conference Location: Kansas City, MO, USA Conference Date: 19920305
Sponsor: ACM
E.I. Conference No.: 17693
Source: SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education) v 24 n 1 Mar 1992. p 163-167
Publication Year: 1992
CODEN: SIGSD3 ISSN: 0097-8418 ISBN: 0-89791-468-6
Language: English
Document Type: JA; (Journal Article) Treatment: A; (Applications); G; (General Review); T; (Theoretical)
Journal Announcement: 9304
Abstract: Computer architecture classes do not provide students with laboratory experience in the design of **instruction set architectures** . Projects that compare designs have not been possible due to a lack of support software. The design and evaluation of a **new instruction set** requires an **assembler** , a symbolic debugger, and a statistics gatherer. Every **new instruction set** requires changes to all three programs. It would be unrealistic to expect that either students or instructor would (re)write such software in order to evaluate each new design. A **new** , flexible software package called the **Instruction Set** Testbed (IST) provides for the comparison of **instruction set architectures** without writing any of the support software. IST's table-driven **assembler** uses a student-supplied architecture definition to **assemble** programs. IST's interactive debugger and a statistics gatherer also have access to the architecture definition. This allows symbolic debugging of the **assembly** language programs and automatic histogramming of instruction usage in the student-defined architecture. IST has been used in both undergraduate and graduate architecture classes to investigate such topics as orthogonality, choice and number of operands, addressing modes, and RISC philosophy. (Author abstract) Refs.
Descriptors: COMPUTER ARCHITECTURE; ENGINEERING EDUCATION; REDUCED **INSTRUCTION SET** COMPUTING; **PROGRAM ASSEMBLERS** ; **PROGRAM** DEBUGGING; STATISTICAL METHODS; COMPUTER SOFTWARE
Identifiers: COMPUTER ARCHITECTURE CLASSES; LABORATORY EXPERIENCE; **INSTRUCTION SET** TESTBED (IST); SOFTWARE PACKAGE IST
Classification Codes:
723 (Computer Software); 901 (Engineering Profession); 922 (Statistical Methods)
72 (COMPUTERS & DATA PROCESSING); 90 (GENERAL ENGINEERING); 92 (ENGINEERING MATHEMATICS)

16/5/7 (Item 7 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

01395249 E.I. Monthly No: EI8310083087 E.I. Yearly No: EI83023347
Title: MU P's ON-CHIP MACROCODE EXTENDS INSTRUCTION SET .
Author: Guttag, Karl
Corporate Source: Texas Instruments Inc, Houston, Tex, USA
Source: Electronic Design v 31 n 5 Mar 3 1983 p 157-161
Publication Year: 1983
CODEN: ELODAW ISSN: 0013-4872
Language: ENGLISH
Journal Announcement: 8310
Abstract: How fast a program can be written and how fast it runs depends largely on the power of a microprocessor's **instruction set** . For that reason, manufacturers are attempting to **upgrade** the basic **command** set. However, the tradeoffs associated with attaching **new instructions** to an existing microprocessor can cut into the device's overall efficiency. By

coding instructions in **assembly** language and storing them as macroroutines in an on-chip ROM, a 16-bit processor can execute floating-point operations and other routines very quickly.
Descriptors: *DATA STORAGE UNITS; COMPUTERS, MICROPROCESSOR
Identifiers: ON-CHIP MACRODES
Classification Codes:
722 (Computer Hardware); 723 (Computer Software)
72 (COMPUTERS & DATA PROCESSING)

16/5/8 (Item 8 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

01067536 E.I. Monthly No: EI8107055251 E.I. Yearly No: EI81018199
Title: MICROCOMPUTER SOFTWARE DEVELOPMENT SYSTEM BASED ON A MINICOMPUTER MACRO- ASSEMBLER .
Author: Dickie, A. A.; Soomro, H. K.
Corporate Source: Univ of Dundee, Scotl
Source: IEEE Transactions on Industrial Electronics and Control Instrumentation v IECI-28 n 1 Feb 1981 p 21-23
Publication Year: 1981
CODEN: IICIAA ISSN: 0018-9421
Language: ENGLISH
Journal Announcement: 8107
Abstract: A description is presented of the use of a general-purpose minicomputer macro- **assembler** and relocatable loader as a technique for **assembling** and loading relocatable programs for various microcomputers. A standard set of library macros are provided to assist the user in constructing a new **assembler** from the minicomputer's macro- **assembler** . The library macros provide syntax- and argument-checking facilities. The microcomputer **assembler** constructed in this way retains all the extended features of the host **assembler** including conditional **assembly** , relocation, and the ability to write further macros in the **new instruction set** . Also, the relocation and interprogram communication facilities of the host **assembler** allow the host loader to be used to construct a final program in the memory of the host machineby linking and loading subroutines and scanning subroutine libraries. 2 refs.
Descriptors: *COMPUTER SOFTWARE; COMPUTERS, MICROPROCESSOR
Classification Codes:
723 (Computer Software); 722 (Computer Hardware)
72 (COMPUTERS & DATA PROCESSING)

16/5/10 (Item 10 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

00687100 E.I. Monthly No: EI7801001252 E.I. Yearly No: EI78015794
Title: COMMON MICROPROCESSOR ASSEMBLY **LANGUAGE.**
Author: Nicoud, J. D.
Corporate Source: Mini & Microcomput Lab, EPFL, Lausanne, Switz
Source: EUROMICRO Symp on Microprocess and Microprogram, 2nd, Venice, Italy, Oct 12-14 1976 Publ by North-Holland Publ Co, New York, NY, 1977 p 213-219
Publication Year: 1976
Language: ENGLISH
Journal Announcement: 7801
Abstract: A set of notations for microprocessor instructions and addressing schemes is proposed, and it is shown by comparing 12 available microprocessors that a common **assembly** language can be defined with the following advantages: much faster learning time for the **instruction set** of a **new** processor, less confusion when changing processors, easier program conversion and benchmark writing. For each processor, a reference card must give the list of the available instructions and their effects on the flags; additional explanations are usually only necessary for a few special instructions since the meaning of the basic instructions never changes. About 60 basic instructions are defined, covering 99 to 100% of

the instructions available in a given processor. 2 refs.
Descriptors: *COMPUTER PROGRAMMING LANGUAGES--*Machine Orientation; COMPUTERS, MICROPROCESSOR
Classification Codes:
723 (Computer Software); 722 (Computer Hardware)
72 (COMPUTERS & DATA PROCESSING)

16/5/11 (Item 11 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

00540670 E.I. Monthly No: EI7605029726 E.I. Yearly No: EI76013304
Title: MESP -- A Generating Aid for User Programs.
Title: MESP, EIN HILFSMITTEL ZUM ERSTELLEN VON ANWENDERPROGRAMMEN.
Author: Schenk, Kurt
Corporate Source: Siemens, Vienna, Austria
Source: Siemens-Zeitschrift v 50 n 2 Feb 1976 p 91-95
Publication Year: 1976
CODEN: SIEZAB ISSN: 0037-4709
Language: GERMAN
Journal Announcement: 7605
Abstract: The MESP standard program for the basic processing of binary and analog process data was developed for process computers not equipped with peripheral storage units. Implementation is in **assembly** language using the **instruction set** of the Siemens 320 process computer with **additional** simple floating-point **instructions** . The concept, functions and list structure of the MESP are described. In German.
Descriptors: *COMPUTER PROGRAMMING; COMPUTER SYSTEMS PROGRAMMING
Classification Codes:
723 (Computer Software)
72 (COMPUTERS & DATA PROCESSING)

16/5/13 (Item 1 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

01237259 ORDER NO: AAD92-25734
A RETARGETABLE COMPILER AND VIRTUAL MACHINE BASED METHODOLOGY FOR ASSESSMENT OF INSTRUCTION SET **ARCHITECTURES**
Author: STEPANIAN, ROBERT
Degree: PH.D.
Year: 1992
Corporate Source/Institution: THE UNIVERSITY OF TEXAS AT AUSTIN (0227)
Supervisor: HARVEY G. CRAGON
Source: VOLUME 53/04-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 1996. 227 PAGES
Descriptors: ENGINEERING, ELECTRONICS AND ELECTRICAL; COMPUTER SCIENCE
Descriptor Codes: 0544; 0984

This research has produced a **new** methodology called ISAS (**Instruction Set** Architecture Assessment System) for performance assessment of proposed and existing **instruction set** architectures, independent of implementation and realization technologies. The methodology supports the configuration of one, two and three-address, scalar SISD (Single Instruction Single Data stream) architectures with a wide array of operation sets, operand access and addressing modes, register set architectures and subroutine linkage mechanisms. After menu-based selection of desired architectural elements from an ISA taxonomy, a compiler, an **assembler** and a virtual machine module, using a standard mnemonic set for the target architecture, are automatically configured. C benchmarks, representing the target environment, can subsequently be compiled, **assembled** and virtually executed to generate a variety of architectural performance measures, including bit budget, bit traffic and number of static/dynamic instructions. The methodology has been validated by comparing measured data from two existing architectures with data obtained by modeling these architectures directly in ISAS. A comprehensive series of

architectural case studies modeled in ISAS additionally demonstrate the applicability and practicality of the methodology in performing efficient architectural assessments.

16/5/14 (Item 2 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

936047 ORDER NO: AAD86-25329
A HORIZONTALLY RECONFIGURABLE ARCHITECTURE FOR EXTENDED PRECISION ARITHMETIC (PARALLEL COMPUTING, CONDITION CODES FACTORING)
Author: CHIARULLI, DONALD MARK
Degree: PH.D.
Year: 1986
Corporate Source/Institution: THE LOUISIANA STATE UNIVERSITY AND AGRICULTURAL AND MECHANICAL COL. (0107)
Source: VOLUME 47/08-B OF DISSERTATION ABSTRACTS INTERNATIONAL. PAGE 3428. 115 PAGES
Descriptors: COMPUTER SCIENCE
Descriptor Codes: 0984

A special computer for high-precision arithmetic and parallel processing which features an ALU that is dynamically reconfigurable under program control has been designed and a prototype machine constructed. The 256-bit ALU consists of eight 32-bit slices each of which has its own ALU operation code in each microinstruction. The slices can remain logically separated from each other, or can be dynamically connected to either or both of their neighbors under control of a segment control code that is part of each microinstruction. The result is a unique parallel architecture which provides real parallelism to user programs at the instruction level while globally retaining a sequential control structure. Management of parallelism is achieved through a two level hierarchy of condition codes and extended **instruction sets** to support conditional **instruction** execution. **New** types of parallel micro-programming tools introduce a system for reconfiguration management and parallel programming. An **assembler** , debug simulator, and interactive operating environment have been implemented. An analysis of the instruction times to execute arithmetic operations on the machine show that it will be exceptionally fast for problems in computational number theory and factoring of integers.

16/5/18 (Item 4 from file: 2)
DIALOG(R)File 2:INSPEC

4975295 INSPEC Abstract Number: B9508-1130B-003, C9508-5210B-002
Title: Synthesis of application specific instruction sets
Author(s): Ing-Jer Huang; Despain, A.M.
Author Affiliation: Inst. of Comput. & Inf. Eng., Nat. Sun Yat-Sen Univ., Kaohsiung, Taiwan
Journal: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems vol.14, no.6 p.663-75
Publication Date: June 1995 Country of Publication: USA
CODEN: ITCSDI ISSN: 0278-0070
U.S. Copyright Clearance Center Code: 0278-0070/95/$04.00
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P); Theoretical (T); Experimental (X)
Abstract: In **instruction set** serves as the interface between hardware and software in a computer system. In an application specific environment, the system performance can be **improved** by designing an **instruction set** that matches the characteristics of hardware and the application. We present a systematic approach to generate application-specific **instruction sets** so that software applications can be efficiently mapped to a given pipelined micro-architecture. The approach synthesizes **instruction sets** from application benchmarks, given a machine model, an objective function, and a set of design constraints. In addition, **assembly** code is generated

to show how the benchmarks can be compiled with the synthesized **instruction set** . The problem of designing **instruction sets** is formulated as a modified scheduling problem. A binary tuple is proposed to model the semantics of instructions and integrate the instruction formation process into the scheduling process. A simulated annealing scheme is used to solve for the schedules. Experiments have shown that the approach is capable of synthesizing powerful instructions for modern pipelined microprocessors, and running with reasonable time and a modest amount of memory for large applications. (21 Refs)
Subfile: B C
Descriptors: computer architecture; **instruction sets** ; logic CAD; pipeline processing; scheduling; simulated annealing
Identifiers: application specific **instruction sets** ; pipelined microarchitecture; application benchmarks; machine model; objective function; design constraints; **assembly** code; synthesized **instruction set** ; modified scheduling problem; binary tuple; semantics modelling; simulated annealing scheme; pipelined microprocessors; **instruction sets** synthesis
Class Codes: B1130B (Computer-aided circuit analysis and design); B0260 (Optimisation techniques); B1265F (Microprocessors and microcomputers); C5210B (Computer-aided logic design); C5220 (Computer architecture); C1180 (Optimisation techniques); C6140B (Machine-oriented languages)

16/5/26 (Item 12 from file: 2)
DIALOG(R)File 2:INSPEC

01081240 INSPEC Abstract Number: C83024069
Title: A systematic approach to the design and implementation of a computer instruction set
Author(s): Marovac, N.
Author Affiliation: Dept. of Math. Sci., San Diego State Univ., San Diego, CA, USA
Journal: Computer Architecture News vol.11, no.1 p.19-24
Publication Date: March 1983 Country of Publication: USA
CODEN: CANED2 ISSN: 0163-5964
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: When selecting the **instruction set** for a computer family, the designer must consider several criteria. The design considerations leading to the final **instruction set** for a **new** computer are very seldom published. The author presents a discussion of the design and implementation criteria for an **instruction set** , and a proposal of a systematic approach for the design and implementation of an **instruction set** . (18 Refs)
Subfile: C
Descriptors: **assembly** language; computer architecture
Identifiers: computer **instruction set** ; design considerations; design and implementation
Class Codes: C5220 (Computer architecture); C6140B (Machine-oriented languages)

16/5/27 (Item 13 from file: 2)
DIALOG(R)File 2:INSPEC

01334031 INSPEC Abstract Number: C79012729
Title: Microcoded scientific instruction set **enhances speed and accuracy of F-Series computers**
Author(s): Geber, C.R.
Journal: Hewlett-Packard Journal vol.29, no.14 p.18-22
Publication Date: Oct. 1978 Country of Publication: USA
CODEN: HPJOAX ISSN: 0018-1153
Language: English Document Type: Journal Paper (JP)
Treatment: General, Review (G); Practical (P)

Abstract: In the **new** F-Series, the scientific **instruction set** has been microcoded, making it part of the computer's standard instruction repertoire and therefore directly accessible by FORTRAN, BASIC, and **assembly** language programs. Microcode implementation, taking advantage of the computational capability of the F-Series' floating point processor, provides dramatic increases in speed and accuracy over the corresponding software routines. (1 Refs)

Subfile: C

Descriptors: general purpose computers

Identifiers: F-Series computers; standard instruction; floating point processor; HP 1000 computer; microcoded scientific **instruction set**

Class Codes: C5420 (Mainframes and minicomputers)

16/5/34 (Item 4 from file: 94)

DIALOG(R)File 94:JICST-EPlus

02656069 JICST ACCESSION NUMBER: 96A0392476 FILE SEGMENT: JICST-E

A Compiler Technique with SIMD Instructions for Multi-media Processing.

SAKAI JUNJI (1); EDAHIRO MASATO (1); KONAGAYA AKIHIKO (1)

(1) NEC Corp.

Denshi Joho Tsushin Gakkai Gijutsu Kenkyu Hokoku(IEIC Technical Report (Institute of Electronics, Information and Communication Enginners), 1996, VOL.95,NO.609(SS95 42-51), PAGE.7-12, FIG.9, TBL.1, REF.7

JOURNAL NUMBER: S0532BBG

UNIVERSAL DECIMAL CLASSIFICATION: 681.3.068 681.32

LANGUAGE: Japanese COUNTRY OF PUBLICATION: Japan

DOCUMENT TYPE: Journal

ARTICLE TYPE: Original paper

MEDIA TYPE: Printed Publication

ABSTRACT: We propose a compiler technique to convert a program written in a high-level language such as C, into an object program that includes SIMD **instructions** . Some recent microprocessors have SIMD **instruction sets** that accelerate multi-media processing. In many cases, however, we have to write a program in **assembly** language to utilize such a SIMD **instruction set** . The method we propose in this paper enables high-level description of SIMD operations by means of unrolling the loop with array elements, rearranging the order of operations and replacing them by SIMD operations. Our prototype implementation proves its effectiveness for some multi-media programs. In this paper we also mention a language extension suitable for SIMD operations. (author abst.)

DESCRIPTORS: compiler; SIMD; multi-media; microprocessor; sequence and arrangement; loop; language design; high level language; language

BROADER DESCRIPTORS: language processor; system program; computer program; software; computer architecture; computer system(architecture); method; information media; arithmetic processor; hardware; closed circuit; path ; subgraph; graph; design; programming language; formal language

CLASSIFICATION CODE(S): JD03040F; JC02010O

16/5/35 (Item 5 from file: 94)

DIALOG(R)File 94:JICST-EPlus

01011448 JICST ACCESSION NUMBER: 90A0377139 FILE SEGMENT: JICST-E

Optimal design of the abstract KL1 instruction set .

HIRANO KIYOSHI (1); GOTO ATSUHIRO (2)

(1) Fujitsu Social Science Labs. Ltd.; (2) Inst. for New Generation Computer Technology

Joho Shori Gakkai Kenkyu Hokoku, 1990, VOL.90,NO.27(ARC-81), PAGE.17-24, FIG.2, REF.4

JOURNAL NUMBER: Z0031BAO ISSN NO: 0919-6072

UNIVERSAL DECIMAL CLASSIFICATION: 681.3.068

LANGUAGE: Japanese COUNTRY OF PUBLICATION: Japan

DOCUMENT TYPE: Journal

ARTICLE TYPE: Original paper

MEDIA TYPE: Printed Publication

ABSTRACT: This paper describes the design of **new** KL1-B: an abstract KL1 **instruction set** . The **new** KL1-B takes memory based execution scheme, i.e., goal arguments are fetched from memory to working registers only when their contents are required for unification, while previous KL1-B fetches all arguments beforehand. By this scheme, a number of goal arguments are not restricted by that of registers, so that physical registers can be used effectively. Also, cost for switching goal contexts can be minimized. In addition, several compile-time optimizations techniques are adopted in the new KL1-B. (author abst.)

DESCRIPTORS: parallel processing; inference; register; speedup; machine language; computer architecture; optimization; program structure; instruction word; knowledge base system; code generation; **assembly** language

BROADER DESCRIPTORS: treatment; modification; improvement; programming language; formal language; language; computer system(architecture); method; structure; **instruction set** ; artificial intelligence system; computer application system; system; compiler; language processor; system program; computer program; software

18/5/1 **(Item 1 from file: 8)**
DIALOG(R)File 8:Ei Compendex(R)

00859962 E.I. Monthly No: EI7911085516 E.I. Yearly No: EI79017845
Title: GENERALIZED CROSS ASSEMBLY **SYSTEM FOR MICRO-COMPUTERS.**
Author: Skordalakis, E.
Corporate Source: Comput Cent, N. R. C. Democritos, Athens, Greece
Source: EUROMICRO Journal (European Association for Microprocessing and Microprogramming) v 5 n 2 Mar 1979 p 82-88
Publication Year: 1979
CODEN: EUJOD4
Language: ENGLISH
Journal Announcement: 7911
Abstract: A methodology is described which permits using the assembly language, the assembler, the linker and the relocating loader of a host computer (hC) for microcomputers. The assembly language of a microcomputer results from the assembly language of the hC by leaving out the computer-dependent **instructions** and defining **new** ones in their place. The assembler, the linker and the relocating loader of the hC are extended in such a way that they can translate these assembly languages for microcomputers. This is accomplished with little effort. Thus, the powerful facilities of large computers are made available to the user of microcomputers. An implementation of the methodology, in a particular host, is reported. 10 refs.
Descriptors: *COMPUTERS, MICROPROCESSOR
Identifiers: MICROCOMPUTERS
Classification Codes:
722 (Computer Hardware)
72 (COMPUTERS & DATA PROCESSING)

18/5/2 **(Item 1 from file: 99)**
DIALOG(R)File 99:Wilson Appl. Sci & Tech Abs

1133988 H.W. WILSON RECORD NUMBER: BAST94002533
PIC microcontroller programmer
Eady, Fred;
Electronics Now v. 65 (Jan. '94) p. 35-41+
DOCUMENT TYPE: Do-it-yourself Work ISSN: 1067-9294 LANGUAGE: English
RECORD STATUS: New record

ABSTRACT: The article introduces a **new** and very popular reduced **instruction** set computer (RISC)-like microcontroller called PIC from Microchip Technology and shows how to build a full-function PIC16C5X microcontroller programmer. These microcontrollers are low-cost, low-power, high speed CMOS devices that contain EPROM. The examples of PIC16C5X hardware and software and a PIC16C5X **cross assembler** will allow builders to develop their own PIC applications. The materials needed to get started cost roughly $70.

```
File 348:EUROPEAN PATENTS  1978-2004/Jan W05
         (c) 2004 European Patent Office
File 349:PCT FULLTEXT  1979-2002/UB=20040129,UT=20040122
         (c) 2004 WIPO/Univentio

Set     Items   Description
S1     423317   ASSEMBL???
S2      68908   (ASSEMBL? OR COMPIL???)(5N)(CODE OR INSTRUCTION? ? OR PROG-
             RAM OR FILE OR DATA OR INFORMATION OR OPERATION? ? OR OPERATO-
             R? ? OR COMMAND? ? OR FUNCTION? ? OR DIRECTIVE? ? OR PROCEDUR-
             E? ?)
S3      57197   INSTRUCTION()SET()ARCHITECTURE? ? OR ISA
S4      26431   (NEW??? OR UPDAT? OR UPGRAD? OR RECENT? OR LATEST OR ADDED
             OR ADDITIONAL OR SUPPLEMENTARY OR EXTRA OR IMPROVED)(5N)(INST-
             RUCTION? ? OR OPERATOR? ? OR OPERAND? ? OR COMMAND? ?)
S5      14334   (OLD?? OR PREVIOUS? OR PRIOR OR PRECEDING OR FORMER? OR OR-
             IGINAL OR INITIAL OR PRIMARY)(5W)(INSTRUCTION? ? OR OPERATOR?
             ? OR OPERAND? ? OR COMMAND? ?)
S6     154820   (TRANSFORM? OR TRANSLAT? OR CONVERT??? OR CONVERSION? ? OR
             CHANG? OR MODIF???? OR MODIFICATION? ? OR AMEND? OR ADJUST???
             OR ADJUSTMENT? ? OR ALTER??? OR ALTERATION? ?)(5N)(INSTRUCTIO-
             N? ? OR OPERATOR? ? OR COMMAND? ? OR CODE OR DATA)
S7        720   S6(5N)(ASCII OR OPCODE OR OP()CODE)
S8        324   CROSS()ASSEMBL??? OR CROSSASSEMBL???
S9         30   S1(S)S7 OR S1(50N)S7
S10        16   S1(50N)S2(50N)S3(50N)S4:S5
S11        14   S1(S)S2(S)S3(S)S4:S5
S12        21   S10:S11
S13      5745   INSTRUCTION()SET
S14        41   S1(S)S2(S)S13(S)S4:S5
S15        58   S1(50N)S2(50N)S13(50N)S4:S5
S16        69   S14:S15
S17        55   S16 NOT (S9 OR S12)
S18         3   S3:S5(50N)S8
```

9/5,K/3 (Item 3 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00872714

Method for path name format conversion
Verfahren zur Formatkonvertierung bei Pfadnamen
Procede de conversion de format des noms d'acces

PATENT ASSIGNEE:
SUN MICROSYSTEMS, INC., (1392737), 901 San Antonio Road, MS PAL01-521, Palo Alto, California 94303, (US), (Proprietor designated states: all)
INVENTOR:
Harper, James M., 1626 Wood Avenue, Colorado Springs, CO 80907, (US)
Berliner, Brian, 379 Silver Spring Circle, Colorado Springs, CO 80919, (US)
LEGAL REPRESENTATIVE:
Hanna, Peter William Derek (72341), Tomkins & Co., 5 Dartmouth Road, Dublin 6, (IE)
PATENT (CC, No, Kind, Date): EP 800142 A1 971008 (Basic)
EP 800142 B1 991124
APPLICATION (CC, No, Date): EP 97200813 970320;
PRIORITY (CC, No, Date): US 626716 960401
DESIGNATED STATES: DE; FR; GB; IT; SE
INTERNATIONAL PATENT CLASS: G06F-009/44; G06F-017/30
CITED PATENTS (EP B): EP 588488 A
CITED REFERENCES (EP B):
ADA LETTERS, vol. 10, no. 1, January 1990 - February 1990, NEW YORK, NY, US, pages 111-117, XP000430618 Y. E. GAIL WANG: "UNIVERSAL FILE NAMES FOR Ada"
1990 IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN, AND CYBERNETICS - CONFERENCE PROCEEDINGS, 4 - 7 November 1990, LOS ANGELES, CA, US, pages 499-504, XP000215394 M. DAVIS ET AL: "Unicode"
IBM TECHNICAL DISCLOSURE BULLETIN, vol. 32, no. 10a, March 1990, ARMONK, NY, US, pages 456-462, XP000083401 "FILE NAME MAPPING IN A HETEROGENEOUS DISTRIBUTED ENVIRONMENT";

ABSTRACT EP 800142 A1
A method and apparatus for converting ASCII path names to parsed path name structures (22B, 32B) provides downward compatibility so that program modules written for modern operating systems which provide parsed path name structure inputs may be run under older operating systems which provide ASCII path name inputs. The method includes, in its most basic form, the steps of converting the prefix and file name of an ASCII path name to a unicode string (22A, 32A), then converting the unicode string to a parsed path structure (22B, 32B). In a preferred embodiment of the invention, the method is implemented in compiled object code written in the "C" computer programming language. The object code defines a parsed path structure, allocates buffers for stack variables, creates various pointers for scanning and counting functions, determines whether or not Uniform Naming Convention is used for the ASCII path name, converts the ASCII code to a unicode string, and then via scanning and counting, converts the first two character spaces in the unicode string to unicode numbers which indicate total string length and prefix length, respectively, and converts every other backslash character within the unicode string to a unicode number which identifies the length of the name component of the string which follows that particular number.
ABSTRACT WORD COUNT: 212
NOTE:
Figure number on first page: 2

LEGAL STATUS (Type, Pub Date, Kind, Text):

Oppn None:	001108	B1	No opposition filed: 20000825
Application:	971008	A1	Published application (Alwith Search Report ;A2without Search Report)
Examination:	980318	A1	Date of filing of request for examination: 980119
Examination:	980506	A1	Date of despatch of first examination report: 980318

Change: 981118 A1 Title of invention (German) (change)
Change: 981118 A1 Title of invention (English) (change)
Change: 981118 A1 Title of invention (French) (change)
Assignee: 990616 A1 Applicant (transfer of rights) (change): SUN MICROSYSTEMS, INC. (1392737) 901 San Antonio Road, MS PAL01-521 Palo Alto, California 94303 (US) (applicant designated states: DE;FR;GB;IT;SE)
Assignee: 990616 A1 Previous applicant in case of transfer of rights (change): SUN MICROSYSTEMS, INC. (1392735) 2550 Garcia Avenue, MS PAL1-521 Mountain View, California 94043-1100 (US) (applicant designated states: DE;FR;GB;IT;SE)
Change: 991124 A1 International Patent Classification changed: 19991001
Grant: 991124 B1 Granted patent
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	9710W1	2722
CLAIMS B	(English)	9947	1310
CLAIMS B	(German)	9947	1186
CLAIMS B	(French)	9947	1556
SPEC A	(English)	9710W1	3516
SPEC B	(English)	9947	3586
Total word count - document A			6240
Total word count - document B			7638
Total word count - documents A + B			13878

...CLAIMS character;
(c) providing for establishing a second buffer within which a parsed path name structure (22B, 32B) corresponding to the ASCII path name will be **assembled** ;
(d) providing for sequentially scanning the ASCII path name, character code value by character code value, beginning with either the first code value of a non-UNC ASCII path name, or a second code value immediately preceding the prefix of a UNC name;
(e) providing for **converting** each character **code** value of the **ASCII** path name as it is scanned to a unicode value;
(f) providing for sequentially writing each unicode value to memory locations within said second buffer...a path element which began with the replaced backslash character unicode value.
16. A computer program product comprising:
a computer usable medium having computer readable **code** embodied therein for **converting** an **ASCII** path name to a parsed path name structure, the computer program product comprising:
(a) computer readable program code devices configured to cause a computer to effect providing the assignment of a buffer within a random access memory in which the parsed path name structure will be **assembled** ;
(b) computer readable program code devices configured to cause a computer to effect providing for converting at least a portion of the ASCII path name...

9/5,K/4 (Item 4 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

00522644
High speed parallel microcode program controller.
Hochgeschwindigkeitsparallelmicrokodeprogrammsteuerung.
Unite de commande a grande vitesse pour programme a microcode parallele.
PATENT ASSIGNEE:
Hughes Aircraft Company, (214913), 7200 Hughes Terrace P.O. Box 45066, Los Angeles, California 90045-0066, (US), (applicant designated states: DE;FR;GB)
INVENTOR:

Coffin, John M., 15226 Ermanita Avenue, Gardena, California 90249, (US)
Seibold, John J., 2107 Grand Avenue No.2, Redondo Beach, California 90278, (US)
LEGAL REPRESENTATIVE:
Witte, Alexander, Dr.-Ing. et al (46523), Witte, Weller, Gahlert & Otten Patentanwalte Augustenstrasse 14, W-7000 Stuttgart 1, (DE)
PATENT (CC, No, Kind, Date): EP 522513 A2 930113 (Basic)
EP 522513 A3 940105
APPLICATION (CC, No, Date): EP 92111522 920707;
PRIORITY (CC, No, Date): US 734343 910709
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: G06F-009/26;
CITED PATENTS (EP A): WO 8401843 A; WO 8401843 A; US 4439827 A; US 4439827 A; US 4587611 A; US 4587611 A; WO 8605015 A; WO 8605015 A; EP 237311 A
CITED REFERENCES (EP A):
MICROPROCESSORS AND MICROSYSTEMS vol. 12, no. 2, March 1988, LONDON GB pages 101 - 106 KOPEC AND LYTLE 'Standalone microsequencer'
32ND IEEE COMPUTER SOCIETY INTERNATIONAL CONFERENCE, COMPCON 87, February 1987, SAN FRANCISCO,US, pages 316 - 321 MCINNIS ET AL. 'VAX 8800 system overview';

ABSTRACT EP 522513 A2
A microprogram controller (50) selects a sequence of low, gate level instructions (microcode) to execute a higher level (assembly language level) instruction. The controller (50) comprises a plurality of accessing means (52, 56, 58, 76, 78, 82) for accessing a microinstruction from a corresponding plurality of different possible blocks (66, 68, 70) of a microprogram store. All accessing means (52, 56, 58, 76, 78, 82) are executed in parallel to provide a plurality of possible microinstructions. Means (64, 72, 74) are provided for selecting which one of the three accessed microinstructions is to be used next in sequence. The parallel nature of this process is fast. A unique instruction set minimizes the number of gates required for microcode storage. A subset of these instructions increases the efficiency of the microcode in terms of both the number of microinstructions required to perform a function and the microcode store address space consumed by those microinstructions. (see image in original document)
ABSTRACT WORD COUNT: 160

LEGAL STATUS (Type, Pub Date, Kind, Text):

Application:	930113 A2	Published application (Alwith Search Report ;A2without Search Report)
Change:	930407 A2	Representative (change)
Search Report:	940105 A3	Separate publication of the European or International search report
Examination:	940831 A2	Date of filing of request for examination: 940628
Withdrawal:	960724 A2	Date on which the European patent application was deemed to be withdrawn: 960201

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF1	686
SPEC A	(English)	EPABF1	4392
Total word count - document A			5078
Total word count - document B			0
Total word count - documents A + B			5078

...SPECIFICATION to be processed in fewer number of clock cycles but would be less efficient in terms of gates.

Typically, microprogram controllers are used to translate **assembly** level instructions into the sequence of events (i.e., the microprogram) required to perform that instruction. **Assembly** level instructions contain an " **op code** " which is **translated** into a microprogram start address. In the controller 50, this translation is performed externally and input via the Start Address Vector bus of the Branch...

...microprogram store mentioned above called the Start UCODE Bloc. This

added block would contain only the first microinstruction of each microprogram. This alternative method allows **assembly** level instructions to be interpreted in fewer clock cycles but is less efficient "gatewise."

The microprogram controller 50 in this exemplary embodiment is controlled by...

9/5,K/6 (Item 6 from file: 348)

00141490

Data processing apparatus and method.
Datenverarbeitungsgerat und -verfahren.
Appareil et procede de traitement d'information.

PATENT ASSIGNEE:
PRIME COMPUTER, INC., (588371), Prime Park, Natick Massachusetts 01760, (US), (applicant designated states: AT;BE;CH;DE;FR;GB;IT;LI;LU;NL;SE)
INVENTOR:
Jones, Walter A., 23 McIntosh Road Chelmsford, Massachusetts 01824, (US)
Jones, Paul R., Jr., 17 Orchard Circle Northboro, Massachusetts 01532, (US)
Ardini, Joseph L., Jr., 86 Sutton Road Needham, Massachusetts 02192, (US)
Papworth, David B., 25 Ledgewood Road Framingham, Massachusetts 01701, (US)
LEGAL REPRESENTATIVE:
Blatchford, William Michael et al (48801), Withers & Rogers 4 Dyer's Buildings Holborn, London EC1N 2JT, (GB)

PATENT (CC, No, Kind, Date): EP 134620 A2 850320 (Basic)
EP 134620 A3 861001
EP 134620 B1 910619
APPLICATION (CC, No, Date): EP 84303073 840508;
PRIORITY (CC, No, Date): PC US 830711
DESIGNATED STATES: AT; BE; CH; DE; FR; GB; IT; LI; LU; NL; SE
INTERNATIONAL PATENT CLASS: G06F-009/38;
CITED PATENTS (EP A): US 3840861 A; US 4025771 A; US 4287561 A; US 3940741 A

CITED REFERENCES (EP A):
P.M. KOGGE: "The Architecture of Pipelined Computers", 1981, pages 231-247, Hemisphere Publishing Corp., London, GB.
THE 8th ANNUAL SYMPOSIUM ON COMPUTER ARCHITECTURE, 12th-14th May 1981, Minneapolis, Minnesota, pages 135-148, IEEE, New York, US; J.E. SMITH: "A study of branch prediction strategies"
IBM TECHNICAL DISCLOSURE BULLETIN, vol. 25, no. 1, June 1982, pages 99-101, New York, US; J.J. LOSQ: "Generalized history table for branch prediction"
IBM TECHNICAL DISCLOSURE BULLETIN, vol. 25, no. 1, June 1982, pages 97-98, New York, US; G.S. RAO: "Technique for minimizing branch delay due to incorrect branch history table predictions"
IBM TECHNICAL DISCLOSURE BULLETIN, vol. 25, no. 4, September 1982, pages 2128-2129, New York, US; L. PARKS et al.: "Target prefetch table";

ABSTRACT EP 134620 A2

Data processing apparatus and method.

A data processing system for processing a sequence of program instructions has two independent pipelines, an instruction pipeline and an execution pipeline. Each pipeline has a plurality of serially operating stages. The instruction stages read instructions from storage and form therefrom address data to be employed by the execution pipeline. The execution pipeline receives the address data and uses it for referencing stored data to be employed for execution of the program instructions. Both pipelines operate synchronously under the control of a pipeline control unit which initiates operation of at least one stage of the execution pipeline prior to completion of the instruction pipeline for a particular instruction. Thereby operation of at least one instruction stage and one execution stage of the respective pipelines overlap for each program instruction. The instruction and execution

pipelines share high speed memory. The pipeline control unit can independently control the flow of instructions through the two pipelines. This is important for operation in conjunction with a microcode storage element which allows conditional branching and subroutine operation. Circuitry also detects pipeline collisions and exception conditions and delays or inhibits operation of one or more of the pipeline stages in response thereto. Under control of the pipeline control unit, one of the independent pipelines can operate while the other is halted. Further, a program instruction flow prediction apparatus and method employ a high speed flow prediction storage element for predicting redirection of program flow prior to the time when the instruction has been decoded. Circuitry is further provided for updating the storage element, correcting erroneous branch and/or non-branch predictions, and accommodating instructions occurring on even or odd boundaries of the normally read double word instruction. Circuitry is further provided for updating the program flow in a single execution cycle so that no disruption to normal instruction sequencing occurs.

ABSTRACT WORD COUNT: 309

LEGAL STATUS (Type, Pub Date, Kind, Text):

Application:	850320	A2	Published application (Alwith Search Report ;A2without Search Report)
Search Report:	861001	A3	Separate publication of the European or International search report
Examination:	870506	A2	Date of filing of request for examination: 870311
Examination:	881012	A2	Date of despatch of first examination report: 880824
Grant:	910619	B1	Granted patent
Lapse:	920115	B1	Date of lapse of the European patent in a Contracting State: SE 910619
Lapse:	920129	B1	Date of lapse of the European patent in a Contracting State: NL 910619, SE 910619
Lapse:	920205	B1	Date of lapse of the European patent in a Contracting State: CH 910619, LI 910619, NL 910619, SE 910619
Lapse:	920408	B1	Date of lapse of the European patent in a Contracting State: AT 910619, CH 910619, LI 910619, NL 910619, SE 910619
Lapse:	920429	B1	Date of lapse of the European patent in a Contracting State: AT 910619, BE 910619, CH 910619, LI 910619, NL 910619, SE 910619
Oppn None:	920610	B1	No opposition filed

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	EPBBF1	3094
CLAIMS B	(German)	EPBBF1	2892
CLAIMS B	(French)	EPBBF1	3708
SPEC B	(English)	EPBBF1	15246
Total word count - document A			0
Total word count - document B			24940
Total word count - documents A + B			24940

...SPECIFICATION ES stage in the case of immediately adjacent instructions. It is possible to make a reasonably accurate determination of what register (if any) will be **modified** by an **instruction** by examining the **opcode** bits and the destination register tag bits of the instruction. Ready access to microcode algorithm related information can be obtained by storing opcode related information in the instruction decode net. Once the microcode for an **assembly** language instruction has been written, a determination is made of the register most likely to be modified by a terminal microcode step. This information is...

9/5,K/14 (Item 8 from file: 349)

00559146 **Image available**

ASSEMBLY LANGUAGE TRANSLATOR

TRADUCTEUR DE LANGAGE D'ASSEMBLAGE

Patent Applicant/Assignee:
ALCATEL USA SOURCING L P,

Inventor(s):
TOWNSEND Arthur R,

Patent and Priority Information (Country, Number, Date):
Patent: WO 200022519 A1 20000420 (WO 0022519)
Application: WO 99US23919 19991014 (PCT/WO US9923919)
Priority Application: US 98173158 19981014

Designated States: AE AL AM AT AT AU AZ BA BB BG BR BY CA CH CN CR CU CZ CZ DE DE DK DK EE EE ES FI FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SK SL TJ TM TR TT UA UG UZ VN YU ZA ZW GH GM KE LS MW SD SL SZ TZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG

Main International Patent Class: G06F-009/44

International Patent Class: G06F-009/45

Publication Language: English

Fulltext Availability:
Detailed Description
Claims

Fulltext Word Count: 6766

English Abstract

A computer-implemented method of translating an assembler program into a high-level language computer program is provided. The method includes receiving each line of the assembler program, parsing individual fields in each assembler program line, including an absolute line number and an opcode for an assembler instruction. Each assembler program line is then stored into a data structure such that each line is accessible and each field in each line is accessible, and the numeric opcode of each assembler program line is parsed into individual digits. Alternatively, the symbolic opcode and operands may be decoded to produce a more human readable and maintainable output. For each assembler program line, a decision tree is traversed in response to the value of each opcode digit to identify the corresponding assembler instruction. The assembler instruction is then translated to an equivalent set of cod e in the high-level computer language. The equivalent set of code for each assembler program line is generated and provided as output.

French Abstract

L'invention concerne un procede configure sur ordinateur de traduction d'un programme d'assemblage dans un programme d'ordinateur a langage evolue. Le procede consiste a recevoir chaque ligne d'un programme d'assemblage, a decouper les champs individuels dans chaque ligne de programme d'assemblage, a inclure un numero de ligne absolu et un code d'operation d'une instruction d'assemblage. Chaque ligne de programme d'assemblage est ensuite memorisee dans une structure de donnees de sorte que chaque ligne reste accessible et que chaque champ dans chaque ligne reste accessible, le code d'operation numerique de chaque ligne de programme d'assemblage etant decoupe en caracteres numeriques individuels. Dans un autre mode de realisation, on peut decoder le code d'operation et les operandes symboliques pour obtenir un produit plus lisible et plus facile a entretenir par l'etre humain. Dans chaque ligne de programme d'assemblage, un arbre de decision est parcouru en reponse a la valeur de chaque caractere numerique de code d'operation, ce qui permet d'identifier l'instruction d'assemblage correspondante. L'instruction d'assemblage est ensuite traduite dans un ensemble de codes equivalent dans le langage machine evolue. L'ensemble de codes equivalent pour chaque ligne de programme d'assemblage est genere et produit.

Fulltext Availability:
Claims

Claim

1 A computer-implemented method of translating an

assembler program into a high-level language computer program, comprising:
receiving each line of the **assembler** program;
parsing individual fields in each **assembler** program line, including an absolute line number and an opcode for an **assembler** instruction;
storing each **assembler** program line into a data structure such that each line is accessible and each field in each line is accessible;
examining the opcode of each **assembler** program line;
for each **assembler** program line, traversing a decision tree in response to value of **opcode** to identify the corresponding **assembler instruction** ;
translating the **assembler instruction** to an equivalent set of code in the high-level computer language; and
outputting the equivalent set of code for each **assembler** program line.
2 The method, as set forth in Claim 1, wherein opcode examining comprises parsing a numeric opcode of each **assembler** program line into individual digits.

3 The method, as set forth in Claim 1, wherein opcode examining comprises:
examining a symbolic opcode; and
examining one...

...in response to the values of the first and second numeric opcode digits;
determining the value of a fourth numeric opcode digit and therefore the **assembler** instruction in response to the values of the first, second and third numeric **opcode** digits; and
translating the **assembler instruction** to an equivalent set of code in the high-level computer language;
outputting the equivalent set of code for each **assembler** program line.

18 The method, as set forth in claim 17, further comprising:
resolving a target address for a jump opcode; and
inserting a unique...

9/5,K/29 (Item 23 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

00106554 **Image available**
DATA PROCESSING SYSTEM
SYSTEME DE TRAITEMENT DE DONNEES
Patent Applicant/Assignee:
INTEL CORP,
Inventor(s):
COLLEY S,
RATTNER J,
COX G,
SWANSON R,
Patent and Priority Information (Country, Number, Date):
Patent: WO 8102477 A1 19810903
Application: WO 80US205 19800228 (PCT/WO US8000205)
Priority Application: WO 80US205 19800228
Designated States: DE GB JP AT CH DE FR GB LU NL SE
Main International Patent Class: G06F-003/00
International Patent Class: G06F-07:00; G06F-09:00; G06F-13:00; G06F-15:16; G06F-15:20
Publication Language: English

Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 139912

English Abstract
A data processor architecture wherein the processors recognize two basic types of objects, an object being a representation of related information maintained in a contiguously addressed set of memory locations. The first type of object contains ordinary data, such as characters, integers, reals, etc. The second type of object contains a list of access descriptors. Each access descriptor provides information for locating and defining the extent of access to an object associated with that access descriptor. The processors recognize complex objects that are combinations of objects of the basic types. One such complex object (94) defines an environment (18 or 20) for execution of objects (92, 93, 98, 106, 122) accessible to a given instance of a procedural operation. The dispatching of tasks to the processor is accomplished by hardware-controlled queuing mechanisms (36), dispatching-port objects (146) which allow multiple sets of processors (38) and (40) to serve multiple, but independent sets of tasks (14, 16). Communication between asychronous tasks or processes is accomplished by related hardware controlled queuing mechanisms (34) (buffered-port objects) (144) which allow messages to move between internal processes or input/output processes without the need for interrupts. A mechanism (42) is provided which allows the processors to communicate with each other. This mechanism is used to reawaken an idle processor to alert the processor to the fact that a ready-to-run process at a dispatching port needs execution.

French Abstract
Structure de processeur de donnees dans laquelle les processeurs reconnaissent deux types fondamentaux d'objets, un objet etant constitue par une representation d'informations connexes maintenues dans un groupe d'emplacements de memoire adresse en contiguite. Le premier type d'objets contient des donnees ordinaires, telles que des caracteres, des nombres entiers, reels, etc. Le deuxieme type d'objets contient une liste de descripteurs d'acces. Chaque descripteur d'acces fournit une information servant a localiser et definir l'etendue de l'acces a un objet associe a ce descripteur. Les processeurs reconnaissent des objets complexes constitues par des combinaisons d'objets des types fondamentaux. Un tel objet complexe (94) definit un environnement (18) ou (20) pour l'execution d'objets (92, 93, 98, 106, 122) accessible a un moment donne d'une operation de traitement. La repartition des taches aux processeurs est executee par des mecanismes (36) de mise en file d'attente commandes par le materiel, des objets (146) de points de connexion de repartition permettant a des groupes multiples de processeurs (38 et 40) d'executer des ensembles de taches (14, 16) multiples mais independantes. La communication entre des taches ou traitement asynchrones est executee par des mecanismes (34) relatifs de mise en file d'attente commandes par le materiel (objets de points de connexion dotes d'un tampon) (144) permettant la circulation des messages entre les traitements internes ou les operations d'entree/sortie sans que des interruptions soient necessaires. Un mecanisme (42) est prevu permettant la communication entre les processeurs. Ce mecanisme est utilise pour reactiver un processeur inactif pour signaler au processeur une operation prete a passer a un point de connexion de repartition avant d'etre executee.

Fulltext Availability:
Detailed Description

Detailed Description
... bit just
read is zero, an access-descriptor-validity fault is generated.

7,5,1,3 Directory Index Search
Given a valid access descriotor, the **conversion** continues by associatively searching the directory-index field of the processor's

12/5,K/2 (Item 2 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01262415
An integer instruction set architecture and implementation
Ganzzahliger-Befehlssatz-Architektur
Architecture de jeu d'instructions de nombres entiers
PATENT ASSIGNEE:
Hitachi, Ltd., (204144), 6, Kanda Surugadai 4-chome, Chiyoda-ku, Tokyo, (JP), (Applicant designated States: all)
INVENTOR:
Krishnan, Sivaram, 1723 Westbrook Avenue, Los Altos, California 94024-5321, (US)
Mandavilli, Srinivas, 162, 4th Main, Vyalikaval, Bangalore-560003, (IN)
Kuo, Liang Cheng, 7562 Hollanderry Place, Cupertino, California 95014, (US)
LEGAL REPRESENTATIVE:
Calderbank, Thomas Roger et al (50122), MEWBURN ELLIS York House 23 Kingsway, London WC2B 6HP, (GB)
PATENT (CC, No, Kind, Date): EP 1089166 A2 010404 (Basic)
APPLICATION (CC, No, Date): EP 308584 000929;
PRIORITY (CC, No, Date): US 410683 991001
DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-009/312; G06F-009/315; G06F-009/305

ABSTRACT EP 1089166 A2
The present invention is directed to a processor element, such as a microprocessor or a micro-controller, structured to execute an integer instruction set architecture. In a specific embodiment, the present invention provides a method for loading an arbitrary constant number into a memory location, through a series of immediate integer instructions. In another specific embodiment present invention provides a method for normalizing a number. The method includes, counting a total number of sign bits in the binary number and then determining a result by subtracting one from the total number. The result may be used for normalizing a number by left-shifting the binary number by the result.
ABSTRACT WORD COUNT: 108
NOTE:
Figure number on first page: 5

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: 010404 A2 Published application without search report
Change: 020403 A2 Inventor information changed: 20020214
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200114	966
SPEC A	(English)	200114	2866
Total word count - document A			3832
Total word count - document B			0
Total word count - documents A + B			3832

...SPECIFICATION using 16 bits, then 32 bits, drove or were in response to electronic circuits 10 using 16 bit, 32 bit, or 64 bit words. Thus **Assembly** Language 16 instruction sets are being designed to handle the increase in application and hardware complexity.
Typical computer programs contain Integer Instructions which perform operations...

...include adding, subtracting, comparing, loading a constant, shifting, moving, logically ORing, or logically NANDing, one or more operands into a result. Some of these integer **operations** were executed by several **Assembly** Language 14 **instructions** . With the complex application programs being more widespread and with wider bus microprocessors, **new instruction set architectures** are needed to take full advantage of

the increased software and hardware complexity.
Therefore, there is a need for an Integer instruction set which makes
...

12/5,K/3 **(Item 3 from file: 348)**
DIALOG(R)File 348:EUROPEAN PATENTS

01045180
Compiler controlled dynamic scheduling of program instructions
Compilergesteuerte dynamische Ablauffolgeplanung von Programmbefehlen
Planification dynamique d'instructions de programme sur commande de compilateur
PATENT ASSIGNEE:
MIPS TECHNOLOGIES INC., (2143720), 600 Mountain Avenue, Murray Hill, New Jersey 07974-0636, (US), (Applicant designated States: all)
INVENTOR:
D'Arcy, Gerard Paul, 114 Deer Run Court, Harleysville, Pennsylvania 19438, (US)
Jinturkar, Sanjay, 3148 East Boulevard, Apt. B, Bethlehem, Pennsylvania 18017, (US)
Glossner, C. John, 226 Alexander Drive, Allentown, Pennsylvania 18104, (US)
Vassiliadis, Stamatis, P. Kemphove 91, Zoetermeer, (NL)
LEGAL REPRESENTATIVE:
Williams, David John et al (86433), Page White & Farrer, 54 Doughty Street, London WC1N 2LS, (GB)
PATENT (CC, No, Kind, Date): EP 924603 A2 990623 (Basic)
EP 924603 A3 010207
APPLICATION (CC, No, Date): EP 98310063 981208;
PRIORITY (CC, No, Date): US 997117 971216
DESIGNATED STATES: DE; FR; GB
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-009/38

ABSTRACT EP 924603 A2
A computer's speed in executing a program is increased based on a compiler controlled dynamic scheduling of program instructions, or an system and method of dynamically storing multiple instruction dependencies in a program. This is accomplished through the use of a single dep instruction, which instructs the computer system's central processing unit executing the program that each instruction associated with the dep instruction can be executed in parallel with one another, subject to the constraints encoded within the dep instruction.
ABSTRACT WORD COUNT: 81
NOTE:
Figure number on first page: 4

LEGAL STATUS (Type, Pub Date, Kind, Text):
Search Report: 010207 A3 Separate publication of the search report
Application: 990623 A2 Published application (Alwith Search Report ;A2without Search Report)
Withdrawal: 030102 A2 Date application deemed withdrawn: 20020702
Examination: 010912 A2 Date of request for examination: 20010720
Change: 010516 A2 Legal representative(s) changed 20010329
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	9925	1854
SPEC A	(English)	9925	7912
Total word count - document A			9766
Total word count - document B			0
Total word count - documents A + B			9766

...SPECIFICATION form of a computer system or machine 1 according to an illustrative embodiment of the present invention. In the figure, a program 2 provides source **code** as input to a compiler/preprocessor 3.

The complier/preprocessor 3 performs both a **compiler function** and a preprocessing **function** in the illustrative embodiment of Fig. 1. However, it will be understood that the **compiler** and preprocessor **functions** can be implemented by separate devices. Moreover, **assembler operations** could be performed by the **compiler** or separately by an assembler (not shown).

The **compiler** /preprocessor 3 examines the source **code** (code is also referred to as instructions within an instruction set architecture (ISA)) and identifies instruction dependencies which can be delimited by a dep instruction (shown in Fig. 2) in order to implement **instruction** level parallelism (ILP). The **compiler** /processor 3 uses a set of optimization rules 4 for this purpose. The **compiler** /preprocessor 3 produces object **code** optimized by the inclusion of dep instructions in order to exploit ILP. Such dep instructions are added as the first instruction of a packet of...

...Fig. 2 as a dep instruction packet 11).

Besides the operation of identifying and implementing the dep instructions, the operations and signals used for the **operations** of the **compiler** /preprocessor 3 are standard. Therefore, the compiler/preprocessor 3 will not be described further. Rather, reference is made to Aho, Compilers, Principles, Techniques and Tools...

12/5,K/4 (Item 4 from file: 348)

00791218

Method and apparatus for transparent emulation of an instruction-set architecture

Verfahren und Anordnung zur transparenten Emulation einer Befehlssatzarchitektur

Procede et dispositif d'emulation transparente d'une architecture de jeu d'instructions

PATENT ASSIGNEE:
International Business Machines Corporation, (200120), Old Orchard Road, Armonk, N.Y. 10504, (US), (Proprietor designated states: all)
INVENTOR:
Emma, Philip George, 28 Fox Den Road, Danbury, Connecticut 06811, (US)
LEGAL REPRESENTATIVE:
Teufel, Fritz, Dipl.-Phys. (11855), IBM Deutschland Informationssysteme GmbH, Patentwesen und Urheberrecht, 70548 Stuttgart, (DE)

PATENT (CC, No, Kind, Date): EP 737914 A1 961016 (Basic)
EP 737914 B1 020612
APPLICATION (CC, No, Date): EP 96103598 960308;
PRIORITY (CC, No, Date): US 421344 950413
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: G06F-009/318
CITED PATENTS (EP B): WO 91/17496 A; WO 94/27214 A; US 5392408 A
CITED REFERENCES (EP B):
BYTE, vol. 19, no. 4, April 1994, PETERBOROUGH, NH, US, pages 119-130, XP000435383 T. R. HALFHILL: "Emulation: RISC's Secret Weapon"
CONFERENCE PROCEEDINGS. 1992 INTERNATIONAL CONFERENCE ON SUPERCOMPUTING, ACM INTERNATIONAL CONFERENCE ON SUPERCOMPUTING, 19 - 23 July 1992, WASHINGTON, DC, US, pages 198-215, XP000576925 G. M. SILBERMAN ET AL: "An Architectural Framework for Migration from CISC to Higher Performance Platforms";

ABSTRACT EP 737914 A1

The invention provides means and methods for extending an instruction-set architecture without impacting the software interface. This circumvents all software compatibility issues, and allows legacy software to benefit from new architectural extensions without recompilation and reassembly. The means employed are a translation engine for translating sequences of old architecture instructions into primary, new architecture instructions, and an extended instruction (EI) cache memory for storing the translations. A processor requesting a

sequence of instructions will look first to the EI-cache for a translation, and if translations are unavailable, will look to a conventional cache memory for the sequence, and finally, if still unavailable, will look to a main memory. (see image in original document)

ABSTRACT WORD COUNT: 134

NOTE:

Figure number on first page: 5

LEGAL STATUS (Type, Pub Date, Kind, Text):

Change:	000719	A1 Legal representative(s) changed 20000531
Examination:	20000315	A1 Date of dispatch of the first examination report: 20000131
Oppn None:	030604	B1 No opposition filed: 20030313
Change:	010523	A1 Title of invention (German) changed: 20010404
Grant:	020612	B1 Granted patent
Application:	961016	A1 Published application (Alwith Search Report ;A2without Search Report)
Examination:	970409	A1 Date of filing of request for examination: 970213

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPAB96	1013
CLAIMS B	(English)	200224	1231
CLAIMS B	(German)	200224	1157
CLAIMS B	(French)	200224	1309
SPEC A	(English)	EPAB96	7072
SPEC B	(English)	200224	7104
Total word count - document A			8086
Total word count - document B			10801
Total word count - documents A + B			18887

...SPECIFICATION binary is loaded into a specific place inmemory where it can be decoded and executed by the processor whose architecture was used to create the **assembly code** .

To a hardware designer, the load module represents the "software interface." That is, the hardware interpretation mechanisms (decoders, etc.) are designed to recognize the binary representations that are in the load module; specifically, the **instruction - set architecture** . Stated more concisely, as seen by the hardware, the **instruction - set architecture** is the software interface.

This interface must be preserved and unchanged no matter what changes are made to the microarchitecture. Next-generation's machines must...

... modules without a hitch. Those load modules contain only those instructions that are part of the architecture today. Enhancing that architecture (i.e., adding **new instructions**) for the next generation machine will not help existing modules; for that matter, since **assemblers** do a rather mundane translation, it will not help existing **assembly** language programs.

When one considers the set of translations between a HLL program, and the final hardware/software interface, a.k.a "target architecture," it ...

...module comprises a sequential set of primitive operations (architected instructions) that implement a more complex operation (high-level-language construct). At the software interface, the **assembly** -language **program** is limited to the expression of those operations that define the architecture.

It has been observed that there are certain fundamental sequences of instructions that...

...SPECIFICATION binary is loaded into a specific place inmemory where it can be decoded and executed by the processor whose architecture was used to create the **assembly code** .

To a hardware designer, the load module represents the "software interface." That is, the hardware interpretation mechanisms (decoders,

etc.) are designed to recognize the binary representations that are in the load module; specifically, the **instruction - set architecture** . Stated more concisely, as seen by the hardware, the **instruction - set architecture** is the software interface.

This interface must be preserved and unchanged no matter what changes are made to the microarchitecture. Next-generation's machines must...

...load modules without a hitch. Those load modules contain only those instructions that are part of the architecture today. Enhancing that architecture (i.e., adding **new instructions**) for the next generation machine will not help existing modules; for that matter, since **assemblers** do a rather mundane translation, it will not help existing **assembly** language programs.

When one considers the set of translations between a HLL program, and the final hardware/software interface, a.k.a "target architecture," it...

...module comprises a sequential set of primitive operations (architected instructions) that implement a more complex operation (high-level-language construct). At the software interface, the **assembly** -language **program** is limited to the expression of those operations that define the architecture.

It has been observed that there are certain fundamental sequences of instructions that...

12/5,K/13 (Item 6 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

00835750 **Image available**

METHOD AND APPARATUS FOR PROCESSOR CODE OPTIMIZATION USING CODE COMPRESSION

PROCEDE ET APPAREIL D'OPTIMISATION DU CODE PROCESSEUR UTILISANT UNE COMPRESSION DE CODE

Patent Applicant/Assignee:
ARC INTERNATIONAL PLC, ARC House - Waterfront Business Park, Elstree Road, Elstree, Herts WD6 3BS, GB, US (Residence), GB (Nationality)
Inventor(s):
[illegible], Peter, 12a Walpole Road, East Ham, London E6 1AR, GB,
Legal Representative:
WARZINSKI Robert F (agent), Gazdzinski & Associates, Suite A232, 3914 Murphy Canyon Road, San Diego, CA 92123, US,

Patent and Priority Information (Country, Number, Date):
Patent: WO 200169376 A2-A3 20010920 (WO 0169376)
Application: WO 2001US8175 20010314 (PCT/WO US0108175)
Priority Application: US 2000189522 20000315
Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR
(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Main International Patent Class: G06F-009/30
Publication Language: English
Filing Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 9508

English Abstract
An improved method of optimizing the instruction set of a digital processor using code compression. In one embodiment, the method comprises obtaining an assembly language program to be used for the optimization process; calculating the static frequency of each instruction type from the base instruction set; sorting the instruction types by frequency; determining the number and type of instructions necessary for correct program execution; creating a compressed instruction set encoding;

re-evaluating the compressed instruction according to the foregoing steps; and generating an instruction set encoding for the compressed instruction set. Improved compressed instruction formats and register structures useful in a processor are also disclosed. A computer program and apparatus for synthesizing logic implementing the aforementioned data cache and pipeline performance enhancements are further disclosed.

French Abstract
Procede ameliore d'optimisation du jeu d'instructions d'un processeur numerique utilisant la compression de code. Dans un mode de realisation, le procede consiste a obtenir un programme de langage d'assemblage a utiliser pour le processus d'optimisation, a calculer la frequence statique de chaque type d'instruction a partir du jeu d'instructions de base, a trier les types d'instruction par frequence, a determiner le nombre et le type d'instructions necessaires a l'execution correcte du programme, a creer un codage du jeu d'instructions comprime, a reevaluer l'instruction comprimee selon les etapes precitees, et a produire un codage du jeu d'instructions pour le jeu d'instructions comprime. L'invention concerne egalement des formats d'instructions comprimes et des structures de registre ameliorees utiles dans un processeur. Un programme et un appareil informatiques de logique de synthese mettant en oeuvre l'architecture d'antememoire de donnees precitee et les ameliorations du fonctionnement en pipeline sont aussi decrits.

Legal Status (Type, Date, Text)
Publication 20010920 A2 Without international search report and to be republished upon receipt of that report.
Search Rpt 20020502 Late publication of international search report
Republication 20020502 A3 With international search report.
Republication 20020502 A3 Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.
Search Rpt 20020502 Late publication of international search report
Correction 20030109 Corrected version of Pamphlet: pages 1/9-9/9, drawings, replaced by new pages 1/11-11/11; due to late transmittal by the receiving Office
Republication 20030109 A3 With international search report.

Fulltext Availability:
Detailed Description

Detailed Description
... result of running the scripts in Appendix 11-VII and following the method of synthesis.

where.
nc = number of compressed instructions
nt = total number of **original** (uncompressed) **instructions**
Appendices VI and VII provide exemplary script to generate a report on analysis of an ARC **assembler file** .

Appendix VIII provides an exemplary script for printing a report for usage of specified processor instruction formats from an **ISA file** and a processor **assembler file** .

Appendices IX and X provide exemplary script to strip out more non-**instruction** lines from an ARC **assembler file** .

Appendix XI provides an exemplary script for moving instructions that are in a branch delay slot to before the branch instruction and removing the delay...

12/5,K/14 **(Item 7 from file: 349)**
DIALOG(R)File 349:PCT FULLTEXT

00828034 **Image available**

AUTOMATED PROCESSOR GENERATION SYSTEM FOR DESIGNING A CONFIGURABLE PROCESSOR AND METHOD FOR THE SAME

SYSTEME AUTOMATISE DE PRODUCTION DE PROCESSEURS, DESTINE A LA CONCEPTION D'UN PROCESSEUR CONFIGURABLE, ET PROCEDE ASSOCIE

Patent Applicant/Assignee:
TENSILICA INC, 3255-6 Scott Boulevard, SantaClara, CA 95054, US, US (Residence), US (Nationality)

Inventor(s):
WANG Albert R, 863 Hunter Lane, Fremont, CA 94539, US,
RUDDELL Richard, 46 Wilder Avenue, Los Gatos, CA 95030, US,
GOODWIN David W, 726 Jackpine Court, Sunnyvale, CA 94086, US,
KILLIAN Earl A, 27961 Central Drive, Los Altos Hills, CA 94022, US,
BHATTACHARYYA Nupur, 855 Park Drive, No. 3, Mountain View, CA 94040, US,
MELINA Marines P, 900 Bayleaf Court, San Jose, CA 95128, US,
LICHTENSTEIN Walter D, 22 Elm Street, Belmont, MA 02478, US,
KONAS Pavlos, 707 Continental Circle #1727, Mountain View, CA 94040, US,
SRINIVASAN Rangarajan, 216 Drakes Bay Avenue, Los Gatos, CA 95032, US,
SONGER Christopher M, 750 N. Shoreline Blvd. #138, Mountain View, CA 94043, US,
PARAMESWAR Akilesh, 3683 Julio Avenue, San Jose, CA 95124, US,
MAYDAN Dror E, 1314 Parkinson Avenue, Palo Alto, CA 94301, US,
GONZALES Ricardo E, 1026 Middle Avenue #D, Menlo Park, CA 94025, US,

Legal Representative:
JOYNER Roger S (et al) (agent), Pillsbury Winthrop LLP, 1600 Tysons Boulevard, McLean, VA 22102, US,

Patent and Priority Information (Country, Number, Date):
Patent: WO 200161576 A2-A3 20010823 (WO 0161576)
Application: WO 2001US5051 20010215 (PCT/WO US0105051)
Priority Application: US 2000506502 20000217

Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR
(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM

Main International Patent Class: G06F-017/50
Publication Language: English
Filing Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 124141

English Abstract

A system for generating processor hardware supports a language for significant extensions to the processor instruction set, where the designer specifies only the semantics of the new instructions and the system generates other logic. The extension language provides for the addition of processor state, including register files, and instructions that operate on that state. The language also provides for new data types to be added to the compiler to represent the state added. It allows separate specification of reference semantics and instruction implementation, and uses this to automate design verification. In addition, the system generates formatted instruction set documentation from the language specification.

French Abstract

L'invention concerne un systeme de production de materiel de traitement, qui supporte un langage destine a des extensions importantes de l'ensemble instructions du processeur et est caracterise en ce que le concepteur specifie seulement la semantique des nouvelles instructions et en ce que le systeme produit une autre logique. Le langage d'extension permet l'ajout d'etat du processeur, notamment de fichiers de registres et d'instructions fonctionnant sur cet etat. Le langage permet encore d'ajouter des nouveaux types de donnees au compilateur -de maniere a representer l'etat ajoute- et de separer la specification de semantique

de reference et l'implementation d'instructions, et il utilise cette fonction pour automatiser la verification de la conception. En outre, ce systeme produit une documentation d'ensembles instructions formates, a partir de la specification de langage.

Legal Status (Type, Date, Text)
Publication 20010823 A2 Without international search report and to be republished upon receipt of that report.
Examination 20011101 Request for preliminary examination prior to end of 19th month from priority date
Search Rpt 20030327 Late publication of international search report
Republication 20030327 A3 With international search report.

Fulltext Availability:
Detailed Description

Detailed Description
... 1 5 With the addition of multi-cycle instructions, it also becomes necessary to generate interlock logic appropriate to the target pipeline for the added **instructions** . This is because with one instruction per cycle issue, no latency one instruction can produce a result that will cause an interlock on the next...

...if their source operands were produced by a two-cycle instruction such as a load). If it is possible to have two-cycle newly-configured **instructions** , there is a need to have following instructions that interlock on the result of the **newly** -configured **instructions** .

Most **instruction set architectures** have multiple implementations for different processor architectures. Prior art.systems combined the specification of the instruction semantics and the implementation logic for instructions and did...

...understanding of what the instruction does. One of the purposes of the reference semantics is to serve as this precise definition. Other components include the **instruction** word, **assembler** syntax, and text description. Prior art systems have sufficient information in the extension language to generate the **instruction** word and **assembler** syntax. With the addition of the reference semantics, only the text description was missing, and there is a need to include the specification of instruction descriptions that can be converted to formatted documentation to produce a conventional **ISA** description book.

Processor development techniques including the above features would render design verification methods of the prior art no longer valid due to their increased...in the configuration environment and, thus, it is shared across all configured sources, developing configurable source code is simplified.

A PERL library for describing the **ISA** has been developed. For TIE, the TIE compiler is run to create the PERL objects for the user-defined **instructions** and this is **added** to the core **ISA** . From there on, all the verification tools query these PERL objects to get the **ISA** and pipeline information of the user-defined TEE.

The following example illustrates how this is done. Starting with a simple TEE description,
opcode acc op2...

...state accum 32
user register 100 accum
iclass acc Jacc) {in ars, in art) (inout accum)
reference acc
assign accum = accum + ars + art;
The TEE **compiler** generates the following **information** about the TIE user state and the semantic of the instruction using it.

State accum mapped to user register: 100, bits 31:0

opcode : acc...

...64
Name : at: input,
regfile : AR, shortname:a, size:32 bits, entries:64
1 5
From the above information, it is possible to generate the **assembly code** for the TIE **instruction** acc. It is known that the instruction has two register operands, both of type AR, based on which it is i le to do some...and using cosimulation to verify the results. These tests use the pipeline specification to exhaustively test all combinations of interlock, bypass, and exceptions.

The HAL **code** generated by the TEE **compiler** is verified by executing it in the **instruction** set simulator. The **assembler** and **compiler** support for the **new instructions** is verified by most of the above.
Cosimulation of Processors
Co-simulation is the process of running the RTL and the reference model in parallel, and 0 comparing the architecturally visible states defined in the **ISA** at specified boundaries.

17/5,K/2 (Item 2 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

1274149
DATA PROCESSOR
DATEN-PROZESSOR
PROCESSEUR DE DONNEES
PATENT ASSIGNEE:
IP Flex Inc., (3265570), 16-6, Ebisunishi 1-chome, Shibuya-ku, Tokyo 150-0021, (JP), (Applicant designated States: all)
INVENTOR:
SATO, Tomoyoshi, 202 Roomy Tsukuba 31 18-10, Higassi 2-chome, Tsukuba-shi Ibaraki 305-0046, (JP)
LEGAL REPRESENTATIVE:
Korber, Martin, Dipl.-Phys. et al (88321), Mitscherlich & Partner Patentanwalte Sonnenstrasse 33, 80331 Munchen, (DE)
PATENT (CC, No, Kind, Date): EP 1215569 A1 020619 (Basic)
WO 200116710 010308
APPLICATION (CC, No, Date): EP 2000956782 000830; WO 2000JP5848 000830
PRIORITY (CC, No, Date): JP 99244137 990830
DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-009/30; G06F-009/38
CITED PATENTS (WO A): JP 7253882 A ; JP 1156824 A ; JP 61294548 A ; JP 2183332 A

ABSTRACT EP 1215569 A1
An instruction set is provided which has a first field for describing an execution instruction for designating content of an operation or data processing that is executed in at least one processing unit forming a data processing system, and a second field for describing preparation information for setting the processing unit to such a state that is ready to execute an operation or data processing that is executed according to the execution instruction, thereby making it possible to provide a control program having the instruction set in which preparation information independent of the execution instruction described in the first field is described in the second field. Accordingly, preparation for execution of the subsequent execution instruction is made based on the preparation information. In the instruction set, since destination of branch instruction is described in the second field and is known in advance, the problems that cannot be solved with a conventional instruction set can be solved.
ABSTRACT WORD COUNT: 157
NOTE:
Figure number on first page: 0001

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: 010502 A1 International application. (Art. 158(1))
Application: 010502 A1 International application entering European phase
Application: 020619 A1 Published application with search report
Examination: 020619 A1 Date of request for examination: 20010403
Assignee: 030910 A1 Transfer of rights to new applicant: IP Flex Inc. (3265571) 27-1, Kamiosaki 2-chome, Shinagawa-ku Tokyo 141-0021 JP
LANGUAGE (Publication,Procedural,Application): English; English; Japanese
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200225	1808
SPEC A	(English)	200225	13406
Total word count - document A			15214
Total word count - document B			0
Total word count - documents A + B			15214

...SPECIFICATION described in the first field, and an instruction or information specifying registers is described in the second field. It

seems be in apparently the same **instruction** set as the conventional assemble code, however, the execution instruction and the preparation information are independent of each other, and therefore are not correspond to...the above method necessarily increases the delay of output. In contrast, the present invention can solve the problem without increasing the delay.

In the instruction **set** of the present invention, it is possible to describe the preparation information prior to the execution **instruction** . Therefore, in a branch instruction such as conditional branch instruction, branch destination information is provided to the control unit **prior** to the execution **instruction** . Namely, in the conventional mnemonic code, a human can understand the whole meaning of the **instruction set** at a glance, but cannot know it until the **instruction set** appears. In contrast, in the **instruction set** of the present invention, the whole meaning of the **instruction set** cannot be understood at a glance, but information associated with the execution instruction are provided before the execution instruction appears. Thus, since the branch destination is assigned **prior** to the execution **instruction** , it is also possible to fetch the **instruction set** at the branch destination, and also to make preparation for the execution instruction at the branch destination in advance.

In general, most of the current...

...pipeline processing speed. Therefore, increasing the processing speed by the prior art cause a significant increase in hardware costs.

As described above, in the conventional **instruction set** , the address information of the branch destination is obtained only after decoding the **instruction set** , making it difficult to essentially solve the penalty produced upon execution of conditional branching. In contrast, in the **instruction set** of the present invention, since the branch destination information is obtained in advance, the penalty produced upon execution of conditional branching is eliminated. Moreover, if...

...preparation is wasted, causing no penalty of the execution time.

Moreover, since the register information required by the subsequent instruction is known simultaneously with or **prior** to the **instruction** execution, the processing speed can be increased without increasing the hardware costs. In the present invention, a part of the processing stage conventionally conducted on the hardware in the conventional pipeline processing is successfully implemented on the software processing in advance during **compiling** or **assembling** stage.

In the **data** processing system of the present invention, the second execution control unit for processing based on the preparation information may be a unit that is capable...

17/5,K/3 (Item 3 from file: 348)

01269949

Method and apparatus for compiling program for parallel processing, and computer readable recording medium recorded with parallelizing compilation program

Verfahren und Gerat zum Ubersetzen eines Programms fur parallelle Verarbeitung, und mit einem parallelisierenden Ubersetzungsprogramm beschriebenes rechnerlesbares Aufzeichnungsmedium

Methode et appareil pour compiler un programme pour traitement parallele, et medium d'enregistrement lisible par ordinateur contenant un programme de compilation parallelisante

PATENT ASSIGNEE:
NEC CORPORATION, (236690), 7-1, Shiba 5-chome, Minato-ku, Tokyo, (JP), (Applicant designated States: all)
INVENTOR:
Obata, Masaya, NEC Corporation, 7-1, Shiba 5-chome, Minato-ku, Tokyo, (JP)
LEGAL REPRESENTATIVE:
VOSSIUS & PARTNER (100314), Siebertstrasse 4, 81675 Munchen, (DE)

PATENT (CC, No, Kind, Date): EP 1094387 A2 010425 (Basic)
APPLICATION (CC, No, Date): EP 122314 001020;
PRIORITY (CC, No, Date): JP 99301316 991022
DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-009/45

ABSTRACT EP 1094387 A2
A parallelizing compilation apparatus for generating object codes that can execute processing, which begin at the branch target where control transfers with higher probability, in advance of the execution of a conditional branch instruction in parallel with the processing prior to the conditional branch instruction without the rearrangement of basic blocks is provided. A branch dualizing section (13) determines, based on profile information (17), the truth probability of the evaluation value of the conditional expression in a conditional branch instruction included in intermediate codes. When the probability of "false" is higher, the branch dualizing section dualizes the conditional branch instruction into a conditional branch instruction whose conditional expression is the inversion of that in the dualized conditional branch instruction and whose branch target is the next instruction of the dualized conditional branch instruction. Conversely, when the probability of "true" is higher, the branch dualizing section inserts an unconditional branch instruction just after the dualized conditional branch instruction and sets the branch target thereof to the next instruction of this unconditional branch instruction. A branch inverting section (19) generates object codes in which the target address of conditional branch instructions and unconditional branch instructions are exchanged, when the determination relating to the truth probability using profile information is inverted with respect to that at the time of the generation of an object code file (16).
ABSTRACT WORD COUNT: 225
NOTE:
Figure number on first page: 1

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: 010425 A2 Published application without search report
Assignee: 030502 A2 Transfer of rights to new applicant: NEC Electronics Corporation (4260580) 1753 Shimonumabe, Nakahara-ku Kawasaki, Kanagawa 211-8668 JP
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200117	1593
SPEC A	(English)	200117	7966
Total word count - document A			9559
Total word count - document B			0
Total word count - documents A + B			9559

...SPECIFICATION field of the third information, into the address corresponding to the label ("L100") indicated by the fourth field of the third information.

More specifically, the **assembler** source program shown in FIG. 8 is converted into that shown in FIG. 9. Conversely, the **assembler** source program shown in FIG. 9 is converted into that shown in FIG. 8. Accordingly, an object code file can be obtained without re-compilation (i.e., processing for generating an **assembler code** from a source **code**) while maintaining performance in **compiling** processing.

As described above, a parallelizing compilation apparatus according to the present embodiment generates object codes that execute processing, which begin at the branch target...

...targets of a conditional branch instruction where the control transfers with higher probability, in advance of the conditional branch instruction in parallel with the processing **prior** to the conditional branch **instruction** . More specifically, the present invention dualizes a

conditional branch instruction to be dualized into a conditional branch instruction and an unconditional branch instruction with reference...

...information on transfer of control at the time of program execution, and inverts the conditional expression in a conditional terminating instruction included in a fork **instruction set** corresponding to the dualized conditional branch instruction, and exchange branch targets of a fork instruction and an unconditional branch instruction.

By the way, the operation of the above-mentioned embodiment has been explained based on the assumption that the object code file 16 shown in FIG. 1 contains **assembler** source programs. However, the parallerizing compiling apparatus may be constructed such that the object code file 16 contains load modules or execution modules and these...

17/5,K/7 (Item 7 from file: 348)

DIALOG(R)File 348:EUROPEAN PATENTS

01081046

Method and apparatus for generating co-simulation and production executables from a single source

Verfahren und Gerat zum Erzeugen von ausfuhrbaren Programmen fur Cosimulation und Produktion aus einer einzelnen quelle

Methode et appareil pour generer des programmes executables de cosimulation et de production a partir d'une source unique

PATENT ASSIGNEE:
Mentor Graphics Corporation, (796271), 8500 S.W. Boeckman Road, Wilsonville, Oregon 97070-7777, (US), (Applicant designated States: all)
INVENTOR:
Finch, Peter M., 12330 SW Alberta, Tigard, Oregon 97223, (US)
LEGAL REPRESENTATIVE:
Wombwell, Francis et al (46021), Potts, Kerr & Co. 15, Hamilton Square, Birkenhead Merseyside L41 6BR, (GB)

PATENT (CC, No, Kind, Date): EP 950967 A2 991020 (Basic)
EP 950967 A3 010808
APPLICATION (CC, No, Date): EP 99300380 990120;
PRIORITY (CC, No, Date): US 58726 980410
DESIGNATED STATES: DE; FR; GB
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-017/50; G06F-011/00

ABSTRACT EP 950967 A2

A storage medium is disclosed. The storage medium having stored on it a set of programming instructions defining a number of data objects and operations on the data objects for use by another set of programming instructions to enable the other set of programming instructions to be compilable into either a version suitable for use in a hardware/software co-simulation that effectively includes calls to hardware simulation functions that operate to generate bus cycles for a hardware simulator, or another version without the effective calls, but explicitly expressed instead, suitable for use on a targeted hardware.

ABSTRACT WORD COUNT: 96

NOTE:
Figure number on first page: 1

LEGAL STATUS (Type, Pub Date, Kind, Text):

Change:	010801	A2	International Patent Classification changed: 20010612
Application:	991020	A2	Published application without search report
Search Report:	010808	A3	Separate publication of the search report
Examination:	011219	A2	Date of request for examination: 20011022

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	9942	937
SPEC A	(English)	9942	3494

Total word count - document A 4431
Total word count - document B 0
Total word count - documents A + B 4431

...SPECIFICATION using an hardware simulator or emulator. Concurrently, the software designer would validate the software using an instruction set simulator on a general purpose computer. The **instruction** set simulator simulates execution of **compiled** **assembly** /machine **code** for determining software correctness and performance at a gross level. These instruction set simulators often include facilitates for handling I/O data streams to simulate to a very limited degree the external hardware of the target design. Typically, **instruction** **set** simulators run at a speeds of ten thousand to several hundred thousand instructions per second, based on their level of detail and the performance of...

...hardware and returns these values to the hardware simulator. Typically, only one to ten instructions per second can be achieved, which is substantially slower than **instruction** **set** simulation.
Recently , increasing amount of research effort in the industry has gone into improving hardware and software co-simulation. New communication approaches such as "message channels" implemented...

...less complete models, such as "bus interface models" for a microprocessor, hardware and software co-simulation known in the art remain running substantially slower than **instruction** **set** simulation.
In U.S. patent application, number 08/645,620, now U.S. Patent 5,xxx,xxx, assigned to the same assignee of the present...

17/5,K/8 (Item 8 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

0812814
A method of producing a computer program
Verfahren zum Erzeugen eines Rechnerprogramms
Methode pour generer un programme d'ordinateur
PATENT ASSIGNEE:
TEXAS INSTRUMENTS INCORPORATED, (279075), 13510 North Central Expressway, Dallas, Texas 75243, (US), (Applicant designated States: all)
INVENTOR:
Davis, Alan L., 3739 Maranatha Drive, Sugar Land, Fort Bend County, Texas 77479, (US)
Humphreys, Jonathan F., 3619 Darnell Ct., Fort Bend County, Texas 77459-4828, (US)
Tatge, Reid E., 1707 Copperwood Lane, Richmond, Harris County, Texas 77469, (US)
LEGAL REPRESENTATIVE:
Holt, Michael (50421), Texas Instruments Limited, P.O. Box 5069, Northampton NN4 7ZE, (GB)
PATENT (CC, No, Kind, Date): EP 833246 A2 980401 (Basic)
EP 833246 A3 991229
APPLICATION (CC, No, Date): EP 97116822 970926;
PRIORITY (CC, No, Date): US 26841 P 960927
DESIGNATED STATES: DE; FR; GB; IT; NL
EXTENDED DESIGNATED STATES: AL; LT; LV; RO; SI
INTERNATIONAL PATENT CLASS: G06F-009/45

ABSTRACT EP 833246 A2
A method of producing a computer program for a computer capable of operating in a plurality of disjoint instruction sets. The method produces a plurality of independently callable functions (202). For each function the method determines a target instruction set employed by the function (203). The method provides the function with a name corresponding to the target instruction set (206). The function name is preferably a modification of a user provided function name corresponding to the target instruction set. The method identifies each call of another independent function and provides each with a name corresponding to the

target instruction set. The method produces a veneer function for each function and for each other instruction set (208). The veneer functions include changing the computer from operating in the other instruction set to operating in the target instruction set, calling the corresponding function, changing the computer to operate in the other instruction set, and a return command. Each veneer function is provided with a name corresponding to the other instruction set (209). Each function and its corresponding veneer functions are converted into a linkable object code module and then linked into an executable object code file of the computer program. The linker preferably omits from the executable object code file any veneer functions not called by a function.

ABSTRACT WORD COUNT: 217

NOTE:

Figure number on first page: 2

LEGAL STATUS (Type, Pub Date, Kind, Text):

Examination:	000823 A2	Date of request for examination: 20000629
Change:	20000105 A2	International Patent Classification changed: 19991113
Application:	980401 A2	Published application (A1with Search Report ;A2without Search Report)
Change:	981230 A2	Representative (change)
Search Report:	991229 A3	Separate publication of the search report

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	9814	622
SPEC A	(English)	9814	4707
Total word count - document A			5329
Total word count - document B			0
Total word count - documents A + B			5329

...SPECIFICATION to specify the target instruction set via a compiler directive included in the source code of the function. In this case, specification of the target **instruction** set takes place only after reading the source code file. The compiler may optionally employ a default instruction set in the absence of a user...a call to the original function in the original instruction set; one or more commands to change the computer from operating in the original instruction **set** of function to operating in the additional **instruction set** ; and a return command in the additional **instruction set** . The assembler then alters the original function name according to the naming convention for the new **instruction set** (block 309). The assembler then stores an object code file of the veneer function (block 310) on disk.

The assembler then checks to determine if veneer functions are needed for another **instruction set** (decision block 311). If there is a need for additional veneer functions for additional instruction sets, then the assembler loops back to block 308 to generate another veneer function. The assembler alters the user specified name according to the naming convention in the new **instruction set** (block 309) and stores a veneer object code file using the altered name (block 310) on disk. The process continues until a veneer function is created, renamed and stored for all **additional instruction** sets (decision block 311). The assembler ends at end block 312.

The foregoing description notes a function naming convention which will now be described in detail. Each function is given a name corresponding to the **instruction** set it employs. In the preferred embodiment the compiler and the **assembler** automatically alters a user specified name to provide the object code file name. In the preferred embodiment, the functions may be written in either ARM...system that uses separately provided object code portions which are not necessarily separately stored. It is possible for plural functions, even functions which employ differing **instruction** sets, to be compiled or assembled into a single object code file. If functions stored within such a combined object code file may be accessed...

...assembler operates in accordance with the description above. Each such function within a combined object code file is named according to the

naming convention. The **compiler** or **assembler** produces a veneer **function** , named according to its **instruction set** , for each **additional instruction set** .

17/5,K/13 **(Item 13 from file: 348)**
DIALOG(R)File 348:EUROPEAN PATENTS
(c) 2004 European Patent Office. All rts. reserv.

00372412
Translation technique
Ubersetzungsverfahren
Methode de traduction
PATENT ASSIGNEE:
AT&T Corp., (589370), 32 Avenue of the Americas, New York, NY 10013-2412, (US), (applicant designated states: DE;FR;GB;IT)
INVENTOR:
Goettelmann, John Charles, 1846 Southeast Drive, Point Pleasant New Jersey 08742, (US)
Hiller, Ronald George, 1 Crawford Place, Old Bridge New Jersey 08857, (US)
Krantzler, Irvan Jay, 17 Alameda Court, Eatontown New Jersey 07724, (US)
Macey, Christopher James, 54 Elm Place, Red Bank New Jersey 07701, (US)
Tuomenoksa, Mark Logan, 20 Frances Street, Shrewsbury New Jersey 07702, (US)
LEGAL REPRESENTATIVE:
Buckley, Christopher Simon Thirsk et al (28912), AT&T (UK) LTD., AT&T Intellectual Property Division, 5 Mornington Road, Woodford Green, Essex IG8 0TU, (GB)
PATENT (CC, No, Kind, Date): EP 372834 A2 900613 (Basic)
EP 372834 A3 920610
EP 372834 B1 960619
APPLICATION (CC, No, Date): EP 89312500 891130;
PRIORITY (CC, No, Date): US 280767 881206
DESIGNATED STATES: DE; FR; GB; IT
INTERNATIONAL PATENT CLASS: G06F-009/455; G06F-009/45;
CITED PATENTS (EP A): EP 168034 A
CITED REFERENCES (EP A):
SIGPLAN NOTICES vol. 22, no. 7, July 1987, YORKTON HEIGHTS N.Y. pages 1 - 13; C. MAY: 'Mimic: A fast system/370 simulator';

ABSTRACT EP 372834 A2
Application programs compiled for a first, "source", computer are translated, from their object form, for execution on a second, "target", computer. The translated application programs are linked or otherwise bound with a translation of the source computer system software. The translated system software operates on the image of the source computer address space in the target computer exactly as it did in the source computer. The semantics of the source computer system software are thus preserved identically. In addition, a virtual hardware environment is provided in the target computer to manage events and to deal with differences in the address space layouts between the source and target computers.
ABSTRACT WORD COUNT: 112

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: 900613 A2 Published application (A1with Search Report ;A2without Search Report)
Search Report: 920610 A3 Separate publication of the European or International search report
Examination: 930127 A2 Date of filing of request for examination: 921130
Examination: 931020 A2 Date of despatch of first examination report: 930908
Assignee: 940622 A2 Applicant (name, address) (change)
Grant: 960619 B1 Granted patent
Oppn None: 970611 B1 No opposition filed
LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	EPAB96	1025
CLAIMS B	(German)	EPAB96	876
CLAIMS B	(French)	EPAB96	1066
SPEC B	(English)	EPAB96	16732
Total word count - document A			0
Total word count - document B			19699
Total word count - documents A + B			19699

...SPECIFICATION on an object code version of the program being translated, it will be appreciated that the input program could be, alternatively, for example, a) an **assembly** language version thereof or b) embodied in so-called interpretive code which is in a form intended to be interpreted on the source machine. An example of the latter are programs output by the UCSD Pascal P- **code compilers** . In some cases, the use of a different, intermediate assembly language may not be necessary. Rather, then expansion can be carried out in terms of the **original assembly** language or other **instruction set** . Moreover, it may be desired to execute the translated **program** in **assembly** form (using an interpreter) rather than compiling it into target machine object form.

It will thus be appreciated that those skilled in the art will...

17/5,K/26 (Item 6 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

.0307016 **Image available**

AN INSTRUCTION SET ARCHITECTURE TO AID CODE GENERATION FOR HARDWARE PLATFORMS MULTIPLE HETEROGENEOUS FUNCTIONAL UNITS

ARCHITECTURE DU JEU D'INSTRUCTIONS DESTINEE A ASSISTER LA GENERATION DE CODE POUR DES PLATEFORMES POSSEDANT DE MULTIPLES UNITES FONCTIONNELLES HETEROGENES

Patent Applicant/Assignee:
PROCELER INC, Building 2, Suite 1730, 2727 Paces Ferry Road, Atlanta, GA 30339, US, US (Residence), US (Nationality)

Inventor(s):
PALEM Krishna, 3226J Post Woods Drive, Atlanta, GA 30339, US,
PATEL Hitesh, Apartment 15E, 4545 River Parkway, Atlanta, GA 30339, US,
YALAMANCHILI Sudhakar, 1726 Barrington Circle, Marietta, GA 30062, US,

Legal Representative:
GOLDMAN William G (agent), Gray Cary Ware & Freidenrich LLP, 1755 Embarcadero Road, Palo Alto, CA 94303-3340, US,

Patent and Priority Information (Country, Number, Date):
Patent: WO 200241104 A2-A3 20020523 (WO 0241104)
Application: WO 2001US43255 20011119 (PCT/WO US0143255)
Priority Application: US 2000715578 20001117

Designated States: AE AL AM AT AU AZ BA BB BG BR BY CA CH CN CR CU CZ DE DK DM EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR
(OA) BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZM ZW
(EA) AM AZ BY KG KZ MD RU TJ TM

Main International Patent Class: G06F-009/45
International Patent Class: G06F-011/32
Publication Language: English
Filing Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 10327

English Abstract
The present invention affords a system and method for simplifying the

development and deployment of high-performance embedded applications on reconfigurable computing systems (100). The invention provides a single, high-level development process, enabling system developers to utilize various programming languages to program both the reconfigurable devices (103-105) and the microprocessor (102) of a reconfigurable computing system.

French Abstract

L'invention concerne un systeme et un procede permettant de simplifier le developpement et le deploiement d'applications integrees de haute performance sur des systemes informatiques reconfigurables. L'invention concerne un processus de developpement unique, de haut niveau, permettant aux developpeurs de systemes d'utiliser divers langages de programmation pour programmer a la fois des dispositifs reconfigurables et le microprocesseur d'un systeme informatique reconfigurable.

Legal Status (Type, Date, Text)

Publication	20020523	A2 Without international search report and to be republished upon receipt of that report.
Search Rpt	20020808	Late publication of international search report pages 1/24-24/24, drawings, replaced by new pages 1/24-24/24; due to late transmittal by the receiving Office
Republication	20020808	A3 With international search report.
Search Rpt	20020808	Late publication of international search report
Examination	20021114	Request for preliminary examination prior to end of 19th month from priority date
Correction	20030213	Corrected version of Pamphlet: pages 1/9-9/9, drawings, replaced by new pages 1/4-4/4; due to late transmittal by the receiving Office
Republication	20030213	A3 With international search report.

Fulltext Availability:
Detailed Description

Detailed Description

... and the lack of portability of applications is, generally, also true of mixed hardware systems.

One conventional reconfigurable computing technique is described in A Dynanuc **Instruction** Set Coinputer, published by M.J. Worthlin and B.L. Hutchings in Proceedings of IEEE Workshop on FPGAs for Custom Computing Machines, at pages 99...

...are dynamically paged by halting the processor and then partially configuring the hardware to include the new instruction. Programs for the processor are written using **assembly** language with each **instruction** having a corresponding **assembly** language opcode. Thus, new **instructions** mean that the **assembly** language needs to be appropriately augmented. The instruction modules used in the DISC processor are implemented a priori.

This processor has a potentially unlimited instruction...

...the main processor complete control over the loading and execution of the reconfigurable hardware configurations. Standard ANSI-C code is used as input to the **compiler** , which generates **code** for the Garp platform. The **compiler** 's target **instruction** set is the MIPS instruction set which includes direct access to the reconfigurable portion of the hardware.

However, the Garp compiler does not compile high-level language statements into **assembly** **code** for execution by the reconfigurable portion of the processor. In addition, the FPGA configuration can only be invoked by using a set of new Garp-specific **instructions** that are unknown to a standard compiler and the programmer must provide **assembly** **code** to interface to the FPGA. There is no means for automatically generating **assembly** **code** to load a configuration, perform register

allocation, execute the configuration, and read a return value from the FPGA.
PRISC, as described by R. Razdan and...

17/5,K/30 **(Item 10 from file: 349)**
DIALOG(R)File 349:PCT FULLTEXT
(c) 2004 WIPO/Univentio. All rts. reserv.

00046266
MICROCONTROLLER INSTRUCTION SET
SEGMENT D'INSTRUCTION POUR MICROCONTROLEUR
Patent Applicant/Assignee:
MICROCHIP TECHNOLOGY INCORPORATED, 2355 West Chandler Boulevard, Chandler, AZ 85224-6199, US, US (Residence), US (Nationality)
Inventor(s):
BOLES Edward Brian, 4220 North Tabor Street, Mesa, AZ 85215, US
DRAKE Rodney, 14825 South 40th Way, Phoenix, AZ 85044, US
JOHANSEN Darrel, 1923 East Belmont Drive, Tempe, AZ 85284, US
MITRA Sumit, 1101 East Warner Road #142, Tempe, AZ 85284, US
TRIECE Joseph W, 15610 South 7th Place, Phoenix, AZ 85048, US
YACH Randy L, 16238 South 25th Street, Phoenix, AZ 85048, US
Legal Representative:
BEARD R William Jr, Frohwitter, Suite 500, Three Riverway, Houston, TX 77056, US
Patent and Priority Information (Country, Number, Date):
Patent: WO 200058828 A1 20001005 (WO 0058828)
Application: WO 2000US7656 20000323 (PCT/WO US0007656)
Priority Application: US 99280112 19990326
Designated States: CN JP KR
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE
Main International Patent Class: G06F-009/30
International Patent Class: G06F-009/34
Publication Language: English
Filing Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 28746

English Abstract
A microcontroller apparatus is provided with an instruction set for manipulating the behavior of the microcontroller. The apparatus and system is provided that enables a linearized address space that makes modular emulation possible. Direct or indirect addressing is possible through register files or data memory. Special function registers, including the Program Counter (PC) and Working Register (W), are mapped in the data memory. An orthogonal (symmetrical) instruction set makes possible any operation on any register using any addressing mode. Consequently, two file registers to be used in some two operand instructions. This allows data to be moved directly between two registers without going through the W register. Thus increasing performance and decreasing program memory usage.

French Abstract
L'invention concerne un microcontroleur equipe d'un segment d'instruction permettant de manoeuvrer le comportement dudit microcontroleur. Le dispositif est concu de maniere a fournir un espace adressable qui rend possible l'emulation des modules. L'adressage direct ou indirect est executable par le biais de piles ou d'une memoire donnees. Des registres de fonctions speciales, comprenant un programme compteur (PC) et un registre de travail (W), sont adresses dans une memoire donnees. Un segment d'instruction orthogonal (symetrique) permet toute operation sur tout registre dans tout mode d'adressage. Par consequent, deux piles peuvent etre utilisees dans deux instructions d'operande. Ce procede permet de deplacer directement des donnees entre deux registres sans qu'il soit pour cela necessaire de passer par le registre de travail (W) . Les performances sont ainsi accrues et l'utilisation de la memoire

programme reduite.

Legal Status (Type, Date, Text)
Publication 20001005 A1 With international search report.
Publication 20001005 A1 Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

Fulltext Availability:
Claims

Claim
... ODtions
n 2's complement number for relative Contents
branch instructions Assigned to
x 'Don't care location (= 'O' or'll') < > Register bit field
The **assembler** will generate **code** with x Opcode Field Descriptions
'0'. It is the recommended form of use
for compatibility with all Microchip soft- Figs 54
ware tools.
d Destination...

...Enable bit
GIEL (INTCON<6>) Figs 55
TBLPTRU Table Pointer (21 -bit)
TBLPTRH Figures 54 and 55 lists the symbols recognized
TBLPTRL by the MPASM **assembler**
TABLAT Table Latch (8-bit) Note 1: Any unused opcode is Reserved.
PRODL Product of Multiply low byte Use of any reserved opcode may
PRODH...2-word instruction, 3 cycles will be executed, 9: If s = '1',
certain registers will be loaded from/into shadow registers. If s = 'O'
no **update** occurs.
Instruction Set Summary
Figs 57
SUBSTITUTE SHEET (RULE 26)
/95
7Not
Mnemonic iscription Cycles 16-bit Opcode Status es
@s
Operands MSb @Sb Aff (cted
SUBFVVB f...

17/5,K/32 **(Item 12 from file: 349)**
DIALOG(R)File 349:PCT FULLTEXT

00574709 **Image available**
METHODS AND APPARATUS FOR SCALABLE INSTRUCTION SET ARCHITECTURE WITH DYNAMIC COMPACT INSTRUCTIONS
PROCEDES ET APPAREIL POUR UNE ARCHITECTURE DE JEU D'INSTRUCTIONS EVOLUTIVE AVEC DES INSTRUCTIONS COMPACTES DYNAMIQUES
Patent Applicant/Assignee:
BOPS INCORPORATED,
Inventor(s):
PECHANEK Gerald G,
BARRY Edwin F,
REVILLA Juan Guillermo,
LARSEN Larry D,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200038082 A1 20000629 (WO 0038082)
Application: WO 99US29516 19991214 (PCT/WO US9929516)
Priority Application: US 98215081 19981218
Designated States: CA CN IL JP KR MX AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE
Main International Patent Class: G06F-015/80
International Patent Class: G06F-009/00
Publication Language: English

Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 13930

English Abstract
A hierarchical instruction set architecture (ISA) (Fig. 1B) provides pluggable instruction set capability and support of array processors (100). A pluggable instruction set can be easily added to a processor architecture for code density and performance enhancements. Addressed herein is a unique compacted instruction set which allows a programmer to dynamically create a set of compacted instructions (510) on a task by task basis. These compacted instructions (510) can be executed on an array processor (100). In addition, the ISA is defined as a hierarchy if ISAs which allows for growth and supports the packing of multiple instructions within a hierarchy of instructions.

French Abstract
L'invention concerne une architecture de jeu d'instructions (ISA) (Fig. B) hierarchique destinee a fournir une capacite de jeu d'instructions enfichable et un appui a des processeurs vectoriels (100). On peut facilement ajouter un jeu d'instructions enfichable a une matrice de processeurs afin d'ameliorer la densite de code et la performance. Dans cette architecture, on peut acceder a un jeu d'instructions compactees unique, afin de permettre a un programmateur de creer de maniere dynamique un ensemble d'instructions compactees (510) sur une tache en fonction des taches, ces instructions compactees pouvant ensuite etre executees sur un processeur vectoriel (100). La structure ISA de cette invention est en outre definie comme une hierarchie de plusieurs structures ISA, ce qui permet une evolution des multiples instructions a l'interieur d'une hierarchie d'instructions, ainsi qu'un appui au tassement de ces multiples instructions.

Fulltext Availability:
Detailed Description

Detailed Description
... code will be encountered. Due to this variance in sequential and parallel code, a different set of instructions may be advantageously selected for a compact **instruction set** implementation in order to better optimize code density for each application. A balanced set of compact instructions may not provide good density and performance characteristics...
...can be run after the ftinctionality of uncompressed code has been proven to create compacted code based upon the actual application. This approach enhances the **primary** purpose for the compacted **instruction set** which is namely to improve code density in the final product **code** . Even though **assembler** programming in compacted **code** is not precluded, it is not anticipated as being a requirement.

1 5 The use of an enhanced tool to support application analysis and the instruction selection process, however, is deemed advantageous.

As can be seen from the **previous** compacted- I **instruction set** discussion, a translation process has been defined which is based upon special separate loadable registers. These registers are required in general one per execution unit...

17/5,K/54 (Item 34 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

00163578 **Image available**
METHOD AND APPARATUS FOR IMPROVED EXECUTION OF INSTRUCTION SEQUENCES
PROCEDE ET APPAREIL D'EXECUTION AMELIOREE DE SEQUENCES D'INSTRUCTIONS
Patent Applicant/Assignee:
KING Ed,

Inventor(s):
KING Ed,
Patent and Priority Information (Country, Number, Date):
Patent: WO 8909962 A1 19891019
Application: WO 89US1393 19890405 (PCT/WO US8901393)
Priority Application: US 88382 19880406
Designated States: AT AT AU BB BE BF BG BJ BR CF CG CH CH CM DE DE DK FI FR GA GB GB HU IT JP KP KR LK LU LU MC MG ML MR MW NL NL NO RO SD SE SE SN SU TD TG
Main International Patent Class: G06F-009/00
International Patent Class: G06F-09:30
Publication Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 8343

English Abstract
The computer system of the present invention provides an improved method and apparatus for executing instruction sets. The present invention utilizes a dual bus architecture to provide greater memory access bandwidth and allowing increased system operation. A high speed access bus is utilized for memory access and a standard bandwidth bus is utilized for third party controllers and peripherals. The processor (17) of the present invention creates a link list of instructions (16). The link list includes the origin and destination of the operation to be executed in the particular instruction step. A ''learn processor'' (18) reviews the link list instruction set and updates pointers to eliminate instruction steps that do not require execution. The learn processor (18) tracks results from instruction sequences, which produce an output. The variable inputs and outputs are stored in a state cache (20). If the variable inputs do not change between one cycle sequence and the next the correct result is simply placed at the proper location in the instruction stream, eliminating the necessity of executing the instruction sequence. By eliminating all unnecessary instructions, the clock cycles per instruction ratio is improved dramatically, leading to more efficient machine operation. The ''learn processor'' (18) also evaluates link list instructions to determine origin points and destination points. Sequential instructions which have identical origin points and destination points may be collapsed to an ''ORG'' point which is, in the preferred embodiment, a link list instruction location which performs a plurality of instructions simultaneously.

French Abstract
Le systeme informatique decrit comprend un procede et un appareil ameliores d'execution d'ensembles d'instructions. Une architecture a double bus permet d'obtenir une bande d'acces a la memoire de plus grande largeur et une exploitation accrue du systeme. Un bus d'acces de haute vitesse est utilise pour acceder a la memoire et un bus a largeur de bande standard est utilise pour des unites de commande independantes et pour des peripheriques. Le processeur (17) genere une liste de liaison d'instructions (16). La liste de liaison comprend l'origine et la destination de l'operation a executer pendant une etape d'instruction particuliere. Un ''processeur d'apprentissage'' (18) revoit l'ensemble des instructions reliees dans la liste et met a jour des pointeurs afin d'eliminer des etapes d'instructions qu'il n'est pas necessaire d'executer. Le processeur d'apprentissage (18) recherche les resultats des sequences d'instructions, qui produisent une sortie. Les entrees et les sorties variables sont enregistrees dans une antememoire d'etat (20). Si les entrees variables ne changent pas entre une sequence de cycles et la sequence suivante, le resultat correct est simplement place a l'endroit approprie dans le courant d'instructions, ce qui rend superflue l'execution de la sequence d'instructions correspondantes. En eliminant toutes les instructions superflues, on ameliore considerablement le taux de cycles d'horloge par instruction, ce qui permet d'exploiter plus efficacement la machine. Le ''processeur d'apprentissage'' (18) evalue egalement les instructions de la liste de liaison afin de determiner leurs points d'origine et de destination. Des instructions sequentielles

ayant des points d'origine et de destination identiques peuvent etre comprimees jusqu'a former un point ''ORG'', en d'autres termes, dans le mode preferentiel de realisation, une adresse dans la liste de liaison d'instructions qui execute simultanement une pluralite d'instructions.

Fulltext Availability:
Detailed Description

Detailed Description
... a limited number of instructions that may be collapsed into a single ORG point, the present invention results in ORG points in place of the **original** 9 **instruction** steps. Because each ORG point may be executed in a single dock cycle, the present invention has reduced the number of clock cycles required to...
...prior art machine to 4 in the present invention.

LANGUAGE PROCESSING
The present invention has application to all types of instruction sets.

A micro code **instruction set** typically begins with a programmer preparing a 20 series of steps in source code (such as fortran, basic etc.). The source code is used to generate "P code" which is an intermediate or source code.

Typically, there are two or three source code instructions for each source
code instruction . A **compiler** may be used to generate templets which generate the same sequences of instructions for a particular type of language 25 function. `Ihe ratio of templet instructions to P code is greater than 1. The
templets are then converted to **assembly** language and from the **assembly**

language, the hardware generates micro code.

```
File 275:Gale Group Computer DB(TM) 1983-2004/Jan 29
         (c) 2004 The Gale Group
File 621:Gale Group New Prod.Annou.(R) 1985-2004/Jan 29
         (c) 2004 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2004/Jan 29
         (c) 2004 The Gale Group
File  16:Gale Group PROMT(R) 1990-2004/Jan 29
         (c) 2004 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 148:Gale Group Trade & Industry DB 1976-2004/Jan 29
         (c)2004 The Gale Group
File 624:McGraw-Hill Publications 1985-2004/Jan 29
         (c) 2004 McGraw-Hill Co. Inc
File  15:ABI/Inform(R) 1971-2004/Jan 29
         (c) 2004 ProQuest Info&Learning
File 647:CMP  Computer Fulltext 1988-2004/Jan W3
         (c) 2004 CMP Media, LLC
File 674:Computer News Fulltext 1989-2004/Jan W4
         (c) 2004 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2004/Jan 15
         (c) 2004 The Dialog Corp.
File 369:New Scientist 1994-2004/Jan W3
         (c) 2004 Reed Business Information Ltd.
File 810:Business Wire 1986-1999/Feb 28
         (c) 1999 Business Wire
File 813:PR Newswire 1987-1999/Apr 30
         (c) 1999 PR Newswire Association Inc
File 610:Business Wire 1999-2004/Jan 29
         (c) 2004 Business Wire.
File 613:PR Newswire 1999-2004/Jan 29
         (c) 2004 PR Newswire Association Inc

Set      Items   Description
S1     1041660   ASSEMBL???
S2      196079   (ASSEMBL? OR COMPIL???)(5N)(CODE OR INSTRUCTION? ? OR PROG-
              RAM OR FILE OR DATA OR INFORMATION OR OPERATION? ? OR OPERATO-
              R? ? OR COMMAND? ? OR FUNCTION? ? OR DIRECTIVE? ? OR PROCEDUR-
              E? ?)
S3       68634   INSTRUCTION()SET()ARCHITECTURE? ? OR ISA
S4      229357   (NEW??? OR UPDAT? OR UPGRAD? OR RECENT? OR LATEST OR ADDED
              OR ADDITIONAL OR SUPPLEMENTARY OR EXTRA OR IMPROVED)(5N)(INST-
              RUCTION? ? OR OPERATOR? ? OR OPERAND? ? OR COMMAND? ?)
S5       30029   (OLD?? OR PREVIOUS? OR PRIOR OR PRECEDING OR FORMER? OR OR-
              IGINAL OR INITIAL OR PRIMARY)(5W)(INSTRUCTION? ? OR OPERATOR?
              ? OR OPERAND? ? OR COMMAND? ?)
S6         576   CROSS()ASSEMBL??? OR CROSSASSEMBL???
S7       51213   INSTRUCTION()SET? ?
S8          24   (S4:S5 OR S7)(S)S6
S9          37   (S4:S5 OR S7)(50N)S6
S10         39   S8:S9
S11         32   RD (unique items)
S12         31   S11 NOT PD>20001222
```

12/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02190326 SUPPLIER NUMBER: 20803296 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Software tools. (Buyers Guide)
UNIX Review's Performance Computing, v16, n7, p9(21)
June 15, 1998
DOCUMENT TYPE: Buyers Guide LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 5863 LINE COUNT: 00565

... Inc. Orbix
Bristol Technology Inc. HyperHelp
Caldera Inc. OpenLinux BASE
Confluent Inc. Visual Thought
CyberSafe Corp. TrustBroker Security SDK
Green Hills Software Inc. Green Hills **Cross Assembler** Development Tools **Instruction Set** Simulator MULTI Builder MULTI Version Control
ICL Inc. Enterprise Software Products DAIS (Object Request Broker)
Kuck & Associates Inc. KAP/Pro Toolset
Melillo Consulting ForExample
Mesa...

12/3,K/2 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01970835 SUPPLIER NUMBER: 18547912
CPU technology has deep roots. (history of microprocessors) (includes related article on sources for more information) (Technology Information)
Gwennap, Linley
Microprocessor Report, v10, n10, p9(6)
August 5, 1996
ISSN: 0899-9341 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 4969 LINE COUNT: 00387

... transistor budget to about 6,000; more complicated control logic enabled the chip to process instructions in as few as five clock cycles. The designers **added** many **new instructions** and redefined the old 8008 opcodes, although 8008 programs could be easily **cross - assembled** to run on the new chip. In all, the 8080 defined 244 of the 256 possible opcodes and included one-, two-, and three-byte instructions...

12/3,K/3 (Item 3 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01781978 SUPPLIER NUMBER: 16931884 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Q & A. (questions-and-answer)
Leichter, Jerrold
Digital Systems Journal, v17, n2, p13(1)
March 13, 1995
ISSN: 1067-7224 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 303 LINE COUNT: 00024

TEXT:
...We are considering upgrading our VAX-11/750 running VMS V4.1 and the RSX emulation package to a new system. My application is a **cross - assembler** compiling code for a Motorola 6809 embedded system. The assembler uses the RSX mode of the VAX. As far as I know, the VAX-emulates 11/750 the RSX **instructions** in hardware, and the **new** system uses software emulation for a Microvax 3100 Model 40. Is such an upgrade viable, or is the RSX mode dead?

12/3,K/4 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01674590 SUPPLIER NUMBER: 15035025 (USE FORMAT 7 OR 9 FOR FULL TEXT)
IOCCC, ASxxxx, MINED, TDE update, and a bug fix.
Volkman, Victor R.
C Users Journal, v12, n3, p117(2)
March, 1994
ISSN: 0898-9788 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 1387 LINE COUNT: 00112

... K&R C compilers may work as well.
ASxxxx Part 3 includes a comprehensive 80-page manual covering functionality provided by all three existing ASxxxx **cross assemblers** and linkers. The documentation lays out the exact specifications of syntax for symbols, labels, assembler directives, and expressions in detail. The manual includes appendices with **instruction set** highlights and supported syntax for Motorola 6800, 6801, 6804, 6805, 68HC08, 6809, 6811 6816, Intel 8080 and 8085, and Zilog Z80 and HD64180 CPUs. [TABULAR...

12/3,K/5 (Item 5 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01611403 SUPPLIER NUMBER: 14038499 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Optimizing compilers improve RISC performance. (Oasys offers code generation for 32-bit RISC)
Williams, Tom
Computer Design, v32, n1, p120(2)
Jan, 1993
ISSN: 0010-4566 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 463 LINE COUNT: 00039

ABSTRACT: Oasys Inc offers a set of optimizing compilers for code generation for 32-bit reduced **instruction set** computing (RISC) and other microprocessors. Six language-specific front ends are supported by the compilers: Ada, C Programming Language, C++, Fortran 77, Fortran 90 and Pascal. The code generation can be done for 13 microprocessors. Prices are $5,800 to $19,000 for compilers, **cross - assembler** and multidebugger. Other features include language-specific lexical analyzer and parser for tree-like program representation, common and advanced global optimizations, and regeneration of code...

12/3,K/6 (Item 6 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01584763 SUPPLIER NUMBER: 13434602 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Microtec Research supports cross development for Intel processors on IBM RS/6000. (Microtec Research Inc.'s microprocessor development software) (Brief Article) (New Products) (Product Announcement)
C Users Journal, v11, n2, p132(1)
Feb, 1993
DOCUMENT TYPE: Product Announcement ISSN: 0898-9788 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 200 LINE COUNT: 00016

TEXT:
...Research, and were ported under an agreement with IBM. On the RS/6000, the cross development tools include an ANSI C cross compiler, a Macro **Cross Assembler** , and the XRAY Debugget. A C++ cross compiler is available for 68000 targets. The XRAY Debugger debugs optimized C code, allowing engineers to work with production-quality code. The XRAY Debugget provides an X Windew System Motif interface. The XRAY Debugger on the RS/6000 supports **instruction set** simulation and debug monitor as

execution environments. With **instruction set** simulation, software development can begin before the target hardware is available, while with monitor-based debugging, developers can perform real-time debugging while software runs...

12/3,K/7 (Item 7 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01532812 SUPPLIER NUMBER: 12579811 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Microtec announces tools for HP Apollo 9000 Series 700 workstations. (Microtec Research Inc.'s XRAY applications programming software) (New Products)(Brief Article) (Product Announcement)
C Users Journal, v10, n9, p127(1)
Sept, 1992
DOCUMENT TYPE: Product Announcement ISSN: 0898-9788 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 122 LINE COUNT: 00010

TEXT:
...Series 700 versions of its Motorola 680x0, Hitachi H8/300, H8/500, and EMICRO microprocessor development tools, ANSI C cross compilers, the XRAY Debugger, Macro **Cross Assemblers** , and linking tools. The XRAY Debugger provides a Motif interface and permits debugging of optimized C code, while the XP, AY command macro language allows automation of complex, repetitive tasks. The XRAY Debugger supports **instruction set** simulation, making software development possible before prototype hardware is available. Microtec C compilers accept both ANSI and K&R C Prices start at $4,300...

12/3,K/8 (Item 8 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01530143 SUPPLIER NUMBER: 12488524 (USE FORMAT 7 OR 9 FOR FULL TEXT)
ANSI C development tools available on HP 9000/700. (American National Standards Institute program development software and compilers from Microtec Inc.)(Brief Article) (New Products) (Product Announcement)
HP Professional, v6, n8, p76(2)
August, 1992
DOCUMENT TYPE: Product Announcement ISSN: 0896-145X LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 133 LINE COUNT: 00011

... RAY Debugger, Optimizing ANSI C Cross Compiler and Macro **Cross Assembler** . The XRAY Debugger on the HP 9000 Series 700 supports **instruction set** simulation as an execution environment, which allows users to develop and debug their software before the prototype hardware is available. I/O and interrupts also...

12/3,K/9 (Item 9 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01358827 SUPPLIER NUMBER: 08216400 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Don't neglect design and development tool support when choosing an MPU. (microprocessor selection by system designers)
Gundry, Bill; Hampton, Tom
Computer Design, v29, n4, p16(2)
Feb 12, 1990
ISSN: 0010-4566 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1462 LINE COUNT: 00120

... 32-bit systems. Most applications in the 4- to 8-bit range benefit from the speed and small code size used in the processor's **instruction**

set . For that reason, the most common tools used with 4- and 8-bit processors are **cross - assemblers** and debuggers.

Design teams should consider two important features when evaluating a **cross - assembler** : host support and relocatable code. Most design teams will develop software for 4- and 8-bit microcontrollers on a host system rather than on the...

...programming language (HLL) such as C or Pascal. Besides the HLL, design teams building applications for 16- and 32-bit processors will probably need a **cross - assembler** , a linker, a librarian, and debugger tools.

One advantage of HLLs is that, for 16- and 32-bit systems, they're portable and easser to use than the processor's own **instruction set** . For design teams hoping to benefit from the wide selection of high-speed 16- and 32-bit processors, an HLL will let them easily port...

12/3,K/10 **(Item 10 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01247734 SUPPLIER NUMBER: 07002919 (USE FORMAT 7 OR 9 FOR FULL TEXT)
An 80386 assembler in Forth: (Forth can be used with a variety of processors.) (details and source code listing)
Dilworth, John B.
Dr. Dobb's Journal of Software Tools, v13, n10, p28(16)
Oct, 1988
LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2932 LINE COUNT: 00220

... This assembler is both a helpful learning aid and a useful tool that is easy to modify or enhance.

The assembler implements the full 80386 **instruction set** , including the privileged instructions available only in protected mode. Thus, it is a complete 80286 and 8086 assembler, since the 80286 **instruction set** is a proper subset of that of he 80386. As written, you can immediately use this assembler in real, non-protected mode with the popular...

...itself is portable and non-processor-specific; because the F83 system has been implemented on 8080, 8086, and 68000 systems, the assembler is also a **cross - assembler** when run on such non-80386 processors.

Forth Assemblers

As with other Forth assemblers, this assembler would normally be used as a supplement to the...

12/3,K/11 **(Item 11 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01164436 SUPPLIER NUMBER: 04321550
8-bit crossassembler **adapts to many** instruction sets . **(Software Review) (evaluation)**
Leibson, Steven H.
EDN, v31, n13, p274(2)
June 26, 1986
DOCUMENT TYPE: evaluation ISSN: 0012-7515 LANGUAGE: ENGLISH
RECORD TYPE: ABSTRACT

8-bit crossassembler **adapts to many** instruction sets . **(Software Review) (evaluation)**

12/3,K/12 **(Item 12 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01143690 SUPPLIER NUMBER: 00633848

Realtime Package for Embedded Micro Ps.
Electronic Engineering Times, n339, p38
July 22, 1985
DOCUMENT TYPE: product announcement ISSN: 0192-1541 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

...ABSTRACT: multitasking programs for embedded microprocessor systems. The Real-Time C package consists of a C programming language compiler and a utility package that includes a **cross assembler** , linker, cross-reference facility, and librarian. The package, which is designed for use with the company's VRTX real-time operating system, allows UNIX programmers to use the standard C interface and recompile for the real-time environment without having to learn a **new** set of I-O **commands** .

12/3,K/13 (Item 13 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01059763 SUPPLIER NUMBER: 00543123
The QWERTY QPAK-68 Development System.
Parker, B.
Call-A.P.P.L.E., v7, n2, p23-27
Feb., 1984
DOCUMENT TYPE: evaluaton ISSN: 8755-4909 LANGUAGE: ENGLISH
RECORD TYPE: ABSTRACT

...ABSTRACT: into a 16-bit machine. The package includes a 68008 board, five well-written manuals, an assembly language reference card, a user's manual, a **cross assembler** and debugger. The 68008 runs at eight MHX, which is eight times faster than the 6502. The 68008 has an identical **instruction set** to the 68000, thus any 68000 programs will run with the 68008. The QPAK-68 is a highly recommended addition to an Apple computer. Sample...

12/3,K/14 (Item 14 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01056484 SUPPLIER NUMBER: 00582021
What's New - 68000 Cross-Assembler.
Byte, v9, n12, p524
Dec., 1984
DOCUMENT TYPE: product announcement ISSN: 0360-5280 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

ABSTRACT: The SX-68 **cross assembler** from Allen Systems for Apple II, II+, and IIe microcomputers is an MC68000 **cross assembler** . It is priced at $100. The **cross assembler** features an editor, fifteen commands, seven pseudo opcodes, and uses the Motorola MC68000 **instruction set** .

12/3,K/15 (Item 1 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2004 The Gale Group. All rts. reserv.

01766784 Supplier Number: 53332233 (USE FORMAT 7 FOR FULLTEXT)
Crossware's New ANSI C Compiler Supports Motorola's Latest ColdFire Chips.
Business Wire, p1086
Dec 2, 1998
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 367

... code into ColdFire compatible assembler code.
The package runs on Windows 95/98/NT4.0 and includes an ANSI C compiler and Support libraries, relocatable **cross assembler** , relocating

linker, library manager and the Crossware's Embedded Development Studio GUI environment.

Printed manuals are included and these are also available on-line allowing details of the ColdFire **instruction set** to be instantly accessed.

The environment allows new users to get started quickly. Programs that will run on the Motorola evaluation boards for the MCF5206...

12/3,K/16 (Item 2 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01091861 Supplier Number: 40575201 (USE FORMAT 7 FOR FULLTEXT)
Macintosh Cross Development Desk Accessory Announced
News Release, p1
Nov 15, 1988
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 483

... a set of ROMulators quick and easy," commented Dave Nohle, chief engineer.

Grammar Engine has also expanded its product line of development tools to include **cross assemblers** , linkers, a "C" language cross development system and EPROM programmers. These **new** products, complete with **instructions** and manuals for generating ROM code, are available for MS-DOS, UNIX/ULTRIX, Macintosh and VAX/VMS computers.

The ROMulator allows in-circuit emulation of...

12/3,K/17 (Item 3 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01069342 Supplier Number: 40339809 (USE FORMAT 7 FOR FULLTEXT)
HARDWARE AND SOFTWARE PRODUCTS ACCELERATED DEVELOPMENT OF DP8344-BASED PERIPHERALS
News Release, pN/A
March 31, 1988
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 398

... programs is included in the MPA.

The DP8344 BCP Assembler System is a new software package which supports product development. The system contains a macro **cross assembler** , line editor, and librarian that are uniquely tailored to the DP8344's **instruction set** .

Introduced last October, the DP8344 lowers the cost of implementing IBM 3270, 3299 and 5250 protocol communications capabilities, and reduces the size and complexity of...

12/3,K/18 (Item 4 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01069328 Supplier Number: 40339794 (USE FORMAT 7 FOR FULLTEXT)
ASSEMBLER SYSTEM SPEEDS CODE DEVELOPMENT FOR DP8344 COMMUNICATIONS PROCESSOR APPLICATIONS
News Release, p1
March 31, 1988

Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 340

... which enables designers to speed development of peripheral and network interface designs based on the DP8344 Biphase Communication Processor. The assembler system contains a Marco **Cross Assembler** , Link Editor and Librarian that are uniquely tailored to the DP8344's **instruction set** .

The Assembler provides nested marco definitions and expansions, source file inclusion, nested conditional assembly, and complete expression syntax. Nested macro definitions and expansions enable the
...

12/3,K/19 **(Item 5 from file: 621)**
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2004 The Gale Group. All rts. reserv.

01034662 Supplier Number: 39983665 (USE FORMAT 7 FOR FULLTEXT)
68000 MICROPROCESSOR TRAINER LINKS TO BBC MICRO
PR Newswire, pN/A
March, 1987
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 279

... code can be assembled. Other features include user defined listing format, full range of pseudo ops, and a help facility which displays the entire 68000 **instruction set** .

The **cross - assembler** package includes a 16kbyte EPROM, two utilities diskettes, and a comprehensive instruction manual. The EPROM is compatible with BBC B, and Master series machines. The...

12/3,K/20 **(Item 6 from file: 621)**
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2004 The Gale Group. All rts. reserv.

01021100 Supplier Number: 39690357 (USE FORMAT 7 FOR FULLTEXT)
SIGNETICS UNVEILS NEXT GENERATION OF HIGH SPEED MICROCONTROLLER FAMILY
PR Newswire, pN/A
Feb 6, 1986
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 649

... the 8X305). "The biggest improvements over the 8X305 are the 8X401's ability to handle interrupts and subroutines in their entirety, as well as a **new instruction set** for more arithmetic and logic operations," said Wayland Ramage, product marketing manager for the 8X401 family.

Ramage added that the 8X401 is supported by a powerful group of development tools-- a symbolic IBM PC-based **cross - assembler** called Fortress, and a prototype development board. Applications notes on the 8X401 are also available.

Harvard Architecture Keys Performance

The key to the chip's...

12/3,K/21 **(Item 7 from file: 621)**

DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2004 The Gale Group. All rts. reserv.

01000053 Supplier Number: 39462170 (USE FORMAT 7 FOR FULLTEXT)
MICROTEC RESEARCH INTRODUCES FOUR MS-DOS CROSS COMPILERS
PR Newswire, pN/A
Jan 1, 1985
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 513

... and
sets with up to 256 elements including SET OF CHAR. Paragon Pascal generates optimized, sharable code that takes full advantage of the target hardware **instruction sets** and memory capabilities.

The Paragon **Cross Assembler** , linker, loader and librarian are included in both C and Pascal Z80 or 8085 compilers packages. Each assembler features a fully manufacturer compatible **instruction set** and directive as well as a powerful macro facility. The linking loader combines independently assembled relocatable object modules, resolves external references and adjusts relocatable addresses...

12/3,K/22 (Item 1 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)
(c) 2004 The Gale Group. All rts. reserv.

[illegible] Supplier Number: 42394167 (USE FORMAT 7 FOR FULLTEXT)
Tool vendors fire up Sparc
Electronic Engineering Times, p70
Sept 30, 1991
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 745

... chip deviates from the standard Sparc architecture in several respects. It has a five-stage pipeline instead of a four-stage pipeline and it adds **extra instructions** for multiply, divide and scan. A generic Sparc compiler will work, but a Sparclite-specific compiler adds a performance edge.

Microtec's Sparclite and generic Sparc tool kits both include an ANSI C optimizing compiler, a **cross assembler** , and the XRAY source-level debugger. They run on IBM PCs as well as SparcStations.

The Microtec tool set addresses embedded concerns by supplying a...

12/3,K/23 (Item 1 from file: 160)
DIALOG(R)File 160:Gale Group PROMT(R)
(c) 1999 The Gale Group. All rts. reserv.

02140356
OASYS ANNOUNCES SOFTWARE DEVELOPMENT TOOLS FOR MOTOROLA'S 88000 RISC FAMILY
News Release February 1, 1989 p. 1

Oasys, Inc. today announced a complete set of software development tools for Motorola's 88000 RISC (reduced **instruction set** computer) microprocessor family. The product, called Oasys 88K Tools, allows software engineers and programmers to develop a range of applications for Motorola's RISC processors...

... products in the development tool set are C, Pascala and Fortran native and cross compilers from Green Hills Software (Glendale, Calif.), and a native and **cross assembler** /linker and debugger from Oasys. All products in the Oasys 88K Tools are fully compliant with the Binary Compatibility Standard (BCS) developed by the 88open...

12/3,K/24 (Item 1 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

12486931 SUPPLIER NUMBER: 64340315 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Coldfire debug.
WILSON, RICHARD
Electronics Weekly, 29
August 2, 2000
ISSN: 0013-5224 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 133 LINE COUNT: 00014

TEXT:
...debugging environments are almost identical. A script file to configure the debugger can be automatically generated by the simulator. The Ansi C compiler and relocateable **cross assembler** have been extended to take advantage of MCF5407 enhancements. These include **additional instructions** and addressing modes.

12/3,K/25 (Item 2 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

07866646 SUPPLIER NUMBER: 16882173 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Motorola Introduces $149.95 DSP56002 Evaluation Module for Turnkey Evaluation and Design.
Business Wire, p5221029
May 22, 1995
LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 848 LINE COUNT: 00082

... a series of evaluation modules for Motorola's DSP products -- is a low-cost platform designed to familiarize customers with Motorola's DSP56002, its architecture, **instruction set** and features. The DSP56002 EVM comes complete with Motorola's DSP56000 **cross assembler** and DSP56002 debug software from Domain Technologies (Plano, TX) running under MS-DOS.
"The EVM delivers the substantial hardware and software development tool performance and...

12/3,K/26 (Item 3 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

04117166 SUPPLIER NUMBER: 07805062 (USE FORMAT 7 OR 9 FOR FULL TEXT)
8- and 16-bit microcontrollers. (Special Report)(includes related article on selecting the best microcontroller) (buyers guide)
Mosley, J.D.
EDN, v34, n20, p108(14)
Sept 28, 1989
DOCUMENT TYPE: buyers guide ISSN: 0012-7515 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 5176 LINE COUNT: 00415

... microcontroller with respect to form, fit, and function.
* How extensive are the mi
crocontroller's softwaredevelopment tools, and will they suit your needs? Is a **cross assembler** available to run on your IBM PC or engineering workstation? How difficult will it be to debug your code? Does the **instruction set** allow compact, efficient, and easy coding?
* Consider the ruggedness of
the microcontroller. Does the chip contain any ESDprotection circuits? Does it have a watchdog timer...

12/3,K/27 (Item 4 from file: 148)

DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

03924948 SUPPLIER NUMBER: 07336142 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Software tools handle all (microprocessor) traffic: universal cross-assemblers. (technical)
Leibson, Steven H.
EDN, v34, n12, p89(7)
June 8, 1989
DOCUMENT TYPE: technical ISSN: 0012-7515 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2693 LINE COUNT: 00218

ABSTRACT: For engineers who routinely work with several types of processors, universal **cross - assemblers** can solve such problems as having to buy a new assembler for each design. Although universal, or retargetable, **cross - assemblers** do not comply precisely with a microprocessor's assembly language, for a wide variety of processors they can translate assembly-language source code into machine code. They also permit engineers to quickly develop subsets of standard processor **instruction sets** and to define one processor's **instruction set** using another processor's mnemonics. Commercially available universal **cross - assemblers** , such as AnyWare Engineering's CASM, are discussed in detail.
... you already own the software-development tools for that processor. Universal cross-assemblers can help you solve these dilemmas permanently.

Although universal (also called retargetable) **cross** -assemblers don't provide exact compliance with a [mu]P's assembly language, they can translate assembly-language source code into machine code for a wide variety of processors. They also let you quickly develop subsets of standard processor **instruction sets** . You might use this capability, for example, to write code for an ASIC processor core you optimized (shrunk) by removing the hardware that executes unnecessary instructions.

You can also use a universal **cross - assembler** to define one processor's **instruction set** using another processor's mnemonics. With this feature, you can convert a program written for one processor into another's machine code rather painlessly.

Despite the multiple benefits of universal **cross assemblers** , vendors that provide assemblers for standard [mu]Ps and [mu]Cs generally don't sell universal **cross - assemblers** . Instead, they offer a range of separate assemblers closely tailored to an individual processor or processor family. For example, Boston Systems Office (Waltham, MA) offers a wide variety of tailored assemblers. It even sells more than one assembler for the Motorola 6800 [mu]P family because the **instruction sets** differ slightly between the 6800 and Hitachi's compatible 6300 [mu]P series. Several other third-party vendors of assemblers for standard processors, including Enertec...classes (sets) of instructions and designate which class the program should use when you assemble a file. You can use this feature to define an **improved** processor's extended **instruction set** . The TASM table for the 6502 [mu]P, for example, defines **extra instructions** for Rockwell's R65C00 and R65C02 microprocessors.

For reasons resembling Anderson's, Jonathan Griffitts wrote a universal **cross - assembler** called CASM to help him with his consulting work. Griffitts has been using his assembler since early 1987 to assemble code for the many different...

...using Borland International's Turbo Pascal compiler. On a friend's recommendation, he transformed Cross-8 into a commercial product.

Cross-8 became the first **cross - assembler** marketed by Aske's company, Universal **Cross - Assemblers** . Because the program was written in Turbo Pascal, versions of Cross-8 were available for Digital Research's CP/M and Microsoft Corp's MS...

...these products assemble code for 16- and 32-bit processors as well as for processors with smaller instruction-word sizes. Both assemblers employ a multipart **instruction - set** table to define the processor instructions. The **instruction - set** tables are stored in text files, so you can use a text editor to modify an existing table or create a new one.

Cross-16 and Cross-32 are written in C, which makes them portable

across a variety of computers. Universal **Cross Assemblers** offers its assemblers for use with the MS-DOS operating system. Macrochip Research, however, licensed the source code for Universal Cross Assemblers' Cross-32 and...code for your [mu]P. You can also use these software-development tools to simplify related design tasks. For example, you can use a universal **cross - assembler** to standardize programs with one assembly-language style or one **instruction set** for every processor you use. This feat is simple to perform with closely related processors, such as the Zilog Z80 and Intel 8085 [mu]Ps...

...8085 [mu]P (load destination from source), for example, you can extend Zilog's format to the Intel processor with the aid of a universal **cross - assembler** . Similarly, if you prefer Intel's syntax, you can extend it to Zilog's processors. You can also use a universal **cross - assembler** to create one assembly-language syntax for several unrelated processors.

Further, you may want to use a particular [mu]P or [mu]C, but discover...

CAPTIONS: Representative universal **cross - assemblers** . (table); Addresses of companies discussed in the article. (table); Microprocessor **instruction sets** supplied with universal **cross - assemblers** . (table)

12/3,K/28 (Item 5 from file: 148)

06512461 SUPPLIER NUMBER: 06512461 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Low-cost software development.

Somers, Jean-Paul

Machine Design, v60, n5, p148(2)
March 10, 1988
ISSN: 0024-9114 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 1058 LINE COUNT: 00081

... as PAGE or TTL appear in the source code, the assemblers treat them as NOPs (no instruction), allowing source programs originally created for more extensive **cross assemblers** to be handled without reediting.

On finding an error, the assemblers generate an error message **preceding** the **instruction** where the error occurred. A message at the end of the program listing gives the total number of detected errors.

On-line assemblers also allow...

12/3,K/29 (Item 6 from file: 148)

01895287 SUPPLIER NUMBER: 02865481 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Introduction to microcomputing. (book reviews)

Gray, Steve

Creative Computing, v9, p245(2)
Aug, 1983
COMMENT TYPE: review ISSN: 0097-8140 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT
WORD COUNT: 237 LINE COUNT: 00020

... of every chapter; later ones include suggestions for programming applications.

The 14 chapters cover an introduction, Number Codes, Nonnumerical Computer Codes, Using the 6800's **Instruction Set** , Introducing the PIA, Aids to Program Writing (algorithms, flowcharts, assembler), Arithmetic Operations, Decisions, Loops, Tables and Lists, Subroutines, Interrupts, Serial I/O, and Introduction to System Configuration. Four appendixes provide the 6800 **instruction set** (86 pages), M6800 **Cross Assembler** Reference Manual (38 pages), answers to some of the exercises (the rest are in the Instructor's Manual), and Some Characteristics of the MC68000 16...

12/3,K/30 (Item 1 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext

00531445 CMP ACCESSION NUMBER: EET19930927S1732
Free stuff for embedded designers (DESKTOP ENGINEERING)
Richard Goering
ELECTRONIC ENGINEERING TIMES, 1993, n 765, 108
PUBLICATION DATE: 930927
JOURNAL CODE: EET LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: Design: Design Automation & Test
WORD COUNT: 538

... board that has a number of assemblers, disassemblers and debugging tools for 8051 family devices. The toll-free number is (800) 451-6644. The company **recently** added a shareware **instruction - set** simulator for the 8051. One slight limitation: The documentation is in German.

You can also download an 8051 monitor/emulator program and an 8051 **cross - assembler** from our own EETnet Engineering Forum. What, not yet a member of EETnet? For information, call (800) 848-8990.

I mentioned something about free hardware...

12/3,K/31 (Item 1 from file: 610)
DIALOG(R)File 610:Business Wire

00314969 20000706188B6567 (USE FORMAT 7 FOR FULLTEXT)
Crossware Adds Debugging Tools to Produce Complete ColdFire Development Suite
Business Wire
Thursday, July 6, 2000 12:19 EDT
JOURNAL CODE: BW LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
DOCUMENT TYPE: NEWSWIRE
WORD COUNT: 375

...a script file to configure the debugger can be automatically generated by
the simulator, further accelerating the development process.

The ANSI C compiler and relocatable **cross assembler** have both been extended to
optionally take advantage of the **new** MCF5407 enhancements. These include **additional instructions** and **additional** addressing modes both of which result
in faster program execution.

```
File 275:Gale Group Computer DB(TM) 1983-2004/Jan 29
         (c) 2004 The Gale Group
File 621:Gale Group New Prod.Annou.(R) 1985-2004/Jan 29
         (c) 2004 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2004/Jan 29
         (c) 2004 The Gale Group
File  16:Gale Group PROMT(R) 1990-2004/Jan 29
         (c) 2004 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 148:Gale Group Trade & Industry DB 1976-2004/Jan 29
         (c)2004 The Gale Group
File 624:McGraw-Hill Publications 1985-2004/Jan 29
         (c) 2004 McGraw-Hill Co. Inc
File  15:ABI/Inform(R) 1971-2004/Jan 29
         (c) 2004 ProQuest Info&Learning
File 647:CMP  Computer Fulltext 1988-2004/Jan W3
         (c) 2004 CMP Media, LLC
File 674:Computer News Fulltext 1989-2004/Jan W4
         (c) 2004 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2004/Jan 15
         (c) 2004 The Dialog Corp.
File 369:New Scientist 1994-2004/Jan W3
         (c) 2004 Reed Business Information Ltd.
File 810:Business Wire 1986-1999/Feb 28
         (c) 1999 Business Wire
File 813:PR Newswire 1987-1999/Apr 30
         (c) 1999 PR Newswire Association Inc
File 610:Business Wire 1999-2004/Jan 29
         (c) 2004 Business Wire.
File 613:PR Newswire 1999-2004/Jan 29
         (c) 2004 PR Newswire Association Inc

Set      Items   Description
S1     1041644   ASSEMBL???
S2      196078   (ASSEMBL? OR COMPIL???)(5N)(CODE OR INSTRUCTION? ? OR PROG-
              RAM OR FILE OR DATA OR INFORMATION OR OPERATION? ? OR OPERATO-
              R? ? OR COMMAND? ? OR FUNCTION? ? OR DIRECTIVE? ? OR PROCEDUR-
              E? ?)
S3       68634   INSTRUCTION()SET()ARCHITECTURE? ? OR ISA
S4      223542   (NEW??? OR UPDAT? OR UPGRAD? OR RECENT? OR LATEST OR ADDED
              OR ADDITIONAL OR SUPPLEMENTARY OR EXTRA)(5N)(INSTRUCTION? ? OR
               OPERATOR? ? OR OPERAND? ? OR COMMAND? ?)
S5       30030   (OLD?? OR PREVIOUS? OR PRIOR OR PRECEDING OR FORMER? OR OR-
              IGINAL OR INITIAL OR PRIMARY)(5W)(INSTRUCTION? ? OR OPERATOR?
              ? OR OPERAND? ? OR COMMAND? ?)
S6      428571   (TRANSFORM? OR TRANSLAT? OR CONVERT??? OR CONVERSION? ? OR
              CHANG? OR MODIF???? OR MODIFICATION? ? OR AMEND? OR ADJUST???
              OR ADJUSTMENT? ? OR ALTER??? OR ALTERATION? ?)(5N)(INSTRUCTIO-
              N? ? OR OPERATOR? ? OR COMMAND? ? OR CODE OR DATA)
S7        1576   S6(5N)(ASCII OR OPCODE OR OP()CODE)
S8         576   CROSS()ASSEMBL??? OR CROSSASSEMBL???
S9          50   S1(S)S7 OR S1(100N)S7
S10         39   RD (unique items)
S11         10   S1(S)S2(S)S3(S)S4
S12         27   S1(100N)S2(100N)S3(100N)S4
S13         28   S11:S12
S14         18   RD (unique items)
S15          0   S1(S)S3(S)S4(S)S5
S16          4   S1(100N)S3(100N)S4(100N)S5
S17      51212   INSTRUCTION()SET? ?
S18          0   S1(S)S17(S)S4(S)S5
S19         23   S1(100N)S17(100N)S4(100N)S5
S20         23   S16 OR S19
S21         11   RD (unique items)
```

10/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01712499 SUPPLIER NUMBER: 16245667 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Philips extends 8051 architecture to 16 bits. (Philips Semiconductors' 8051XA-1 microcontroller) (includes related article on price and availability) (Product Announcement)
Wenman, Linley
Microprocessor Report, v8, n13, p17(3)
Oct 3, 1994
DOCUMENT TYPE: Product Announcement ISSN: 0899-9341 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 1863 LINE COUNT: 00141

... mapping from 8051 to XA instructions, there should be few (if any) performance issues when moving to the new architecture. Furthermore, programmers familiar with 8051 **assembly** language can easily program the new processor.
The drawback to the Philips approach is that old binaries will not run on the new processor. Because...

...old binaries to new binaries, and Philips provides a translator to do just that. Unfortunately, the length of some 8051 instructions was increased to make **opcode** space for the new XA **instructions** , so the **translated** binary will expand slightly, changing the instruction addresses. The translator handles simple cases, like branches, but some sequences--jump tables, for example must be converted...

10/3,K/2 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01679328 SUPPLIER NUMBER: 15313308 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Multimedia mail: new multimedia messaging standards and innovations mean more power and better interoperability for enterprise E-mail. (Interoperability: Practical Solutions to Complex Networks special section) (includes related article on E-mail systems that support the Multipurpose Internet Mail Extensions standard)
Freed, Ned
LAN Magazine, v9, n5, pS29(8)
May, 1994
ISSN: 0898-0012 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 5044 LINE COUNT: 00392

... the ASCII character set. And like all RFC 822 messages, the content of MIME messages can be limited to relatively short line of seven-bit **ASCII** . Nontext **data** , then, has to be **converted** to **ASCII** characters of seven-bit bytes before then can be transmitted over the network. This conversion is usually done automatically.
If all types of mail messages...

...to describe different message types and different multimedia body parts, so the mail system can properly interpret each part of the e-mail message and **assemble** and present the pieces correctly.
The MIME specification defines four different header fields:
* A MIME-Version header, which labels a message as MIME-confornant. This...

10/3,K/3 (Item 3 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01602272 SUPPLIER NUMBER: 13924819 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Developer's guide to FoxPro add-on tools. (add-on software for the FoxPro

database management system) (Directory)
Garcia, AnnMarie
Data Based Advisor, v11, n6, p77(9)
June, 1993
DOCUMENT TYPE: Directory ISSN: 0740-5200 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3286 LINE COUNT: 00279

... C library that creates indexes for memo field, enabling contents of memo files to be word searched. Price: $99 single-user; $200 multi-user
Cryptor
Assembler library that provides data encryption by intercepting the data that moves between disk and FoxPro. Price: $99 single-user; $299 multi-user 6 pack; runtimes...

...DOS, useful for high volume printing in single-user environment. Price: $49
Tape2DBF
C library PLB and controller card for 9-track tape units; direct **conversion** of mini/mainframe **data** into FoxPro tables with EBCDIC/ **ASCII** **code** **conversion** , all controlled from FoxPro functions. Price: $1,500

10/3,K/4 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01585125 SUPPLIER NUMBER: 13420462 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Binary translation. (running the binary code of DEC's VAX and MIPS architectures on the company's new Alpha AXP computers) (one of four articles on DEC's Alpha architecture) (contains related articles on terminology used in the main article, the design of the DECchip 21064 microprocessor and the technology used in the 21064 microprocessor) (Cover Story)
Sites, Richard L.; Chernoff, Anton; Kirk, Matthew B.; Marks, Maurice P.; Robinson, Scott G.
Communications of the ACM, v36, n2, p69(15)
Feb, 1993
DOCUMENT TYPE: Cover Story ISSN: 0001-0782 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 9042 LINE COUNT: 00745

... basic blocks, and creates an extended basic block, which includes pseudo-opcodes that indicate the MIPS code idiom.
After the optimizer completes the list of **instructions** , it **translates** each abstract **opcode** to zero or more Alpha AXP opcodes, again building a linked list of instructions. This process may permit further improvements, so the optimizer makes a...
...temporary registers. The register assigner will load and spill MIPS resources and generate temporary registers as needed.
Finally, the list of Alpha AXP instructions is **assembled** into a binary stream, and the instruction scheduler rearranges them to remove resource latencies and use the chip's multiple-issue capability.
Image Formats
The...

10/3,K/5 (Item 5 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01549002 SUPPLIER NUMBER: 12959191 (USE FORMAT 7 OR 9 FOR FULL TEXT)
The best all-around. (Software Review) (Micro Focus Cobol, one of seven articles on Cobol compilers, special section on Cobol compilers) (Cover Story) (Evaluation)
Weston, Rusty; Eliot, Lance B.
Corporate Computing, v1, n6, p98(2)
Dec, 1992

DOCUMENT TYPE: Evaluation ISSN: 1065-8610 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 763 LINE COUNT: 00060

... easily check syntax on our code and move between the various parts of the tool set. A data file editor handles various file formats and **data** structures. You can **convert** between **ASCII** and EBCDIC. The debugger provides an impressive animation capability.

The documentation--though only satisfactory--was the best we found in a group of products that...

...compiler) when they are at the tail end of the development cycle.

Micro Focus also provides a CICS emulation capability, an IMS emulator, a 370 **assembler** emulation feature, and several other IBM mainframe-related tools (such as an EBCDIC-to-ASCII conversion utility). In addition, Micro Focus offers a standard programmer...

10/3,K/6 (Item 6 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01448627 SUPPLIER NUMBER: 11237921 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Share and share alike. (Software Review) (Metaview 2.1 programming tool) (evaluation)
Davis, Tom
MIDRANGE Systems, v4, n14, p34(2)
July 9, 1991
DOCUMENT TYPE: evaluation ISSN: 1041-8237 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 931 LINE COUNT: 00070

...ABSTRACT: an AS/400. Metaview 2.1 has excellent image processing capabilities, and can utilize programmable workstation-microcomputer systems and a central host. The software automatically **converts** EBCDIC **data** on the AS/400 to **ASCII** code for the IBM PC to use. The Metaview-userprog interface can be used to issue commands on the AS/400. Data can be retrieved from the AS/400, processed, and returned back to a host program. A C or **assembler** programmer is not needed to obtain results. The program is well-documented. Prices are $2,400; a $500 runtime program is needed for each workstation...

10/3,K/7 (Item 7 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01313007 SUPPLIER NUMBER: 07892584 (USE FORMAT 7 OR 9 FOR FULL TEXT)
IBM Programmable Network Access. (IBM's data conversion software package)
Computergram International, n1308, CGI11160014
Nov 16, 1989
ISSN: 0268-716X LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 1167 LINE COUNT: 00096

TEXT:

...Dat Link Control. In the second release, the PS/2 with the software and co-processor fitted can serve as an X25 concentrator and packet **assembler** -disassembler and offers protocol and **data** stream **conversions** for **ASCII** terminals, enabling users of those terminals to access SNA or X25 applications more easily. There is also an open, programmable interface so that the product...

... to 64Kbps on the upstream link. Programmable Network Access provides access to SNA applications designed to support 3270 terminals, such as CICS and IMS, from **ASCII** terminals, supported by protocol and **data** stream **conversions** between the terminals and the network. Terminal emulation services provide IBM 3278-2 screen appearances and keyboard functions equivalent to ASCII terminals. Terminals supported through...

...an X25 PSDN; data transfers between locally attached X25 terminals through Programmable Network Access; concentration of ASCII terminal data streams through a CCITT 3X Packet **Assembler** -Disassembler built into the program; SNA host links for X25 terminals through XI and NPSI; and concentration of all X25 downstream traffic on a single...

10/3,K/8 **(Item 8 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01312807 SUPPLIER NUMBER: 07763320 (USE FORMAT 7 OR 9 FOR FULL TEXT)
1STCLASS and COURIERS make binary transfers easy. (includes related article on the script language used in 1STCLASS)
Maclean, Pete
PC Magazine, v8, n19, p399(7)
Nov 14, 1989
ISSN: 0888-8507 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3678 LINE COUNT: 00277

... create or read messages, and you must set up and manage a subdirectory structure. 1STCLASS does automate the MCI connection procedure, however, and it does **translate** binary **data** into **ASCII** format so you can send any PC file as an attachment to a message. Moreover, 1STCLASS is compatible with Express, so users of either program...

...can be downloaded in executable form from PC MagNet, as described in the sidebar "1STCLASS by Modem." Alternatively, you can download the source code and **assemble** it with a Microsoft or IBM macro **assembler** (Version 2 or later). The BASIC files are also available via PC MagNet; if run in BASIC, these will produce the same .COM files. If...

10/3,K/9 **(Item 9 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01282639 SUPPLIER NUMBER: 07881632
New IBM software lets users blend asynch ASCII traffic with SDLC: future version will integrate disparate data on X.25 links.
Smith, Tom
Network World, v6, n45, p6(1)
Nov 13, 1989
ISSN: 0887-7661 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

...ABSTRACT: which will enable users to integrate asynchronous ASCII traffic into asynchronous Systems Network Architecture (SNA) backbone networks. Version 1.0 runs on PS-2s and **converts** asynchronous **ASCII data** into a synchronous EBCDIC data stream for transmission over a single line. Release 1.1 comes bundled with a packet **assembler** -disassembler for transporting SNA, native X.25 and ASCII data over a single X.25 connection. Both packages are scheduled for 1990 release. The packages...

10/3,K/10 **(Item 10 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01261998 SUPPLIER NUMBER: 07197865 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Utilities and languages. (buyers guide)
DG Review, v6, n2, p4(7)
Winter, 1988
DOCUMENT TYPE: buyers guide ISSN: 1050-9127 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 9439 LINE COUNT: 00836

... readers allowing manual and optimized blocking for single and multiple volume labeled and unlabeled custom data storage, transport and

retrieval; storage, communications and IRS; and **data conversion** tools, including **ASCII** to Hollerith zoned decimal and math converters. Retail management is under development. Products are fully documented and many are available in source code. Call or...

...AOS/DVS, MS-DOS, CP/M, CP/M-86 Languages: PASCAL, ALGOL, PL/1, FORTRAN IV, 5, 77, Business BASIC, Extended BASIC, COBOL, Interactive COBOL, **Assembly** Reference Number: 888 Contact: Steven G. Walter Telephone: (803) 288-3060
DATA GENERAL CORP. SOFTWARE PRODUCTS AND SERVICES DIVISION
Application: Tape management system Product Name...

10/3,K/11 (Item 11 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01252518 SUPPLIER NUMBER: 06833377 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Article finder. (index) (EDGE Report)
Electronic Design, v36, n15, p93(45)
June, 1988
ISSN: 0013-4872 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 50668 LINE COUNT: 04386

... will prevail, libraries of modules to grow .. E Electronic Design 1/7/88 p96; 1 pp Boston Systems Office ... C toolkit for 68030 includes compiler, **assembler** , linker, and librarian...B Electronic Design 1/21/88 p206 BV Engineering...(LCFIL) Low-cost ($100) powerful filter synthesis and analysis software program designs and analyzes configurations of LC filters...E Microwaves & RF March '88 p227; 2 pp Design...(8085) **Assembly** -language program divides 32-bit by 16-bit numbers...E EDN 3/3/88 p174; 1 pp Development Associates...(Future86) Language for PC produces very...

...pp Hem Data ... (Snap-Filter) Digital-filtering software for data-acquisition program simulates FIR and HR filters...B Electronic Design 2/18/88 p179 Hem **Data** ...(Universal **Translator**) **Translator converts** PC **data** among **ASCII** , floating point, hex, condensed hex, and binary ... B Electronic Design 3/17/88 p149 Hunter Systems...(XDOS) Binary compiler translates PC code for workstations, eliminates...

...MatrixPlotter) Utility program rasterizes complex images for output on a dot-matrix printer...B Electronic Design 2/18/88 p179 Intermetrics...(InterTools) Cross compilers, cross **assemblers** among time-saving software for developing microprocessor systems ... A Electronic Design 2/4/88 p59, 3/3/88 p24 Math Works...(Matlab) Interactive scientific and...

10/3,K/12 (Item 12 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01251535 SUPPLIER NUMBER: 06827781 (USE FORMAT 7 OR 9 FOR FULL TEXT)
FastCAD. (Software Review) (one of 12 CADD packages evaluations) (evaluation)
Schleter, Will
PC Magazine, v7, n14, p169(3)
Aug, 1988
DOCUMENT TYPE: evaluation ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1034 LINE COUNT: 00082

... expandability are definite strengths of FastCAD. Internally, the system is driven entirely by text commands. In fact, the standard menu file is just a simple **ASCII** text file containing FastCAD **commands** and can be **modified** at will. For automatic demonstrations, presentations, and tutorials the script file capability is great, but a learn mode for creating the script files is glaringly...

...to-use macro facility allows even more customization. Perhaps the most powerful feature is the capability for third-party developers to add extended procedures, an **assembly** -language interface method for adding new commands and entities to FastCAD.

Even with its limitations, working with FastCAD is a pure joy. Its user interface...

10/3,K/13 **(Item 13 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01246002 SUPPLIER NUMBER: 06829705 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Utilities and languages. (supplement to DG Review) (directory)
DG Review, v8, n10, pS4(8)
June, 1988
DOCUMENT TYPE: directory ISSN: 1050-9127 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 9309 LINE COUNT: 00835

... readers allowing manual and optimized blocking for single and multiple volume labeled and unlabeled custom data storage, transport and retrieval; storage, communications and IRS; and **data conversion** tools, including **ASCII** to Hollerith zoned decimal and math converters. Retail management is under development. Products are fully documented and many are available in source code. Call or...

...AOS/DVS, MS-DOS, CP/M, CP/M-86 Languages: PASCAL, ALGOL, PL/1, FORTRAN IV, 5, 77, Business BASIC, Extended BASIC, COBOL, Interactive COBOL, **Assembly** Reference Number: 888 Contact: Steven G. Walter Telephone: (803) 288-3060

Concept Automation, Inc. Application: Application access menu Product Name: menuOP Description: menuOP is a...

10/3,K/14 **(Item 14 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01209281 SUPPLIER NUMBER: 06143322 (USE FORMAT 7 OR 9 FOR FULL TEXT)
PC tutor.
Hummel, Robert L.
PC Magazine, v6, n19, p527(4)
Nov 10, 1987
ISSN: 0888-8507 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 1008 LINE COUNT: 00075

... the keyboard with any key you select.

When a key is pressed, the keyboard generates a unique scan code that identifies the key. The BIOS **translates** the scan **code** into the more familiar **ASCII** code. A few keys (the shift keys alone, for example) have no ASCII equivalents or, like SYS REQ, are ignored by the BIOS.KEYSUB cannot be used for such keys.

You can create KEYSUB.COM either from the **assembly** language listing in Figure 1 or by running the BASIC program shown in Figure 2. If you choose the **assembler** route, use Version 2.0 (or later) of the IBM or Microsoft **assembler** and enter the following commands: MASH KEYSUB; LININK KEYSUB; EXE2BIN KEYSUB KEYSUB.COM Ignore the "no stack segment" warning displayed by the linker: it's...

10/3,K/15 **(Item 15 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01207700 SUPPLIER NUMBER: 06168660 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Software: Lotus-compatible products. (Listings)

Bowmer, Katherine
Lotus, v3, n6, p152(5)
June, 1987
ISSN: 8756-7334 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 2560 LINE COUNT: 00206

... data files to PC-DOS data files and automatically provides EBCDIC to ASCII conversion, fixed-field formatting, and conversion from fixed to variable length records. **Assembly** -language programs transfer 1,000 records of 200 bytes in length in less than one minute, and programs can be run in batch mode. Transfers...

...on the program needed for the customer's system.
Format-1-2-3
Reliable Software, P.O. Box D, Titusville, FL 32781, 305-267-2043.
Converts any **ASCII** **data** into 1-2-3 or Symphony worksheet files by interactive or batch processing. Includes communications feature for sending Lotus spreadsheets over telephone lines. Works with...

10/3,K/16 (Item 16 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01205828 SUPPLIER NUMBER: 04655439 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Pixel alignment of EGA fonts. (programming practices)
Cockerham, John T.
PC Tech Journal, v5, n1, p165(12)
Jan, 1987
ISSN: 0738-0194 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2795 LINE COUNT: 00203

... location. This is not a problem from a code standpoint, because well-behaved 8086 machine code is fully relocatable. However, of in-line tables are **assembled** into an external subprogram for its use, the subprogram has no direct means of addressing the in-line tables.
The trick employed by EGAFONT to...

...caller's DS and BP values are pushed onto the stack, a CALL is made to dummy procedure dost2. Because dost2 immediately follows the CALL **opcode** , no significant **change** occurs in the order that **instructions** are executed. In executing the CAM, the CPU pushes the return address onto the stack. This address, which is a 16-bit offset into the code segment, corresponds to ale label dost3 and is immediately popped off the stack into AX. By subtracting the **assembler** -generated value of the label dost3 (which is the offset of dost3 from the first byte of generated code in the external subprogram) from the...

...first byte of the subprogram into the code segment is generated and stored on the stack in variable csx. By adding this value to the **assembler** -generated offset of the in-line tables, the true address of the tables can be generated a runtime.
Font information is maintained in a font...

10/3,K/17 (Item 17 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01168147 SUPPLIER NUMBER: 04322057
Diskoverer. (a disk sector editor program)
Gardner, Bob
Nibble, v7, n8, p15(28)
Aug, 1986
ISSN: 0734-3795 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

ABSTRACT: Diskoverer is a disk editor program providing the ability to modify a disk, to **change** HEX or **ASCII** **data** , to inspect file format or

disk, to select formats for display, to easily locate a file, and to dump a screen to the printer. Diskoverer is easy to use and is fast. The program offers several options for displaying and editing. Diskoverer is written in **assembly** language for efficiency. A program listing is provided, and the program is also available on disk.

10/3,K/18 (Item 18 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01102831 SUPPLIER NUMBER: 00535491 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Assembly Language: The Nature of DOS.
Lafore, R.
PC Magazine, v3, n5, p185-205
March 20, 1984
ISSN: 0888-8507 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 9291 LINE COUNT: 00662

... pretty familiar in the program in Figure 5, except for this one:
08F1:0109 DB 'Good Morning, Robert!$' This doesn't look like an ordinary **assembly** language instruction, and it's not. In fact, it's a very strange sort of animal. Instead of being an instruction that tells the 8088 microprocessor to do something, it's an instruction that tells DEBUG (or the **assembler** program) what to do. In this case, it tells DEBUG to put into memory all the bytes represented by the characters between the single quotation marks. Thus G is **translated** into its **ASCII code** 47h, o into 6Fh, and so on. These values are then placed in memory. Note that DB itself is

10/3,K/19 (Item 19 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01102828 SUPPLIER NUMBER: 00535486 (USE FORMAT 7 OR 9 FOR FULL TEXT)
PC Program Editors: The Next Generation.
Hallings, S.
PC Magazine, v3, n5, p148-156
March 20, 1984
DOCUMENT TYPE: evaluaton ISSN: 0888-8507 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 5162 LINE COUNT: 00385

... TED has a unique extra touch. It can automatically enter the time, date, and version number each time it saves a file.
We read an **Assembler** program, which had been created elsewhere, into TED with the INSERT FILE command. The program appeared on the screen with a non-ASCII tab character...

...each tab spacing, so that the label, operator, and operands on a line all ran together. We got back the tab spacings by rereading the **Assembler** program with the INSERT FILE command. The tests using the COBOL program and the dBASE II command file ran without problems. TED had no trouble...

...II database records. The full record appeared on the screen, where it was broken into 75-character lines with continuation characters. Since TED claims to **translate** non- **ASCII data** into IBM display symbols, we gave it an additional test. We attempted to edit another dBASE II database record that had not been ASCII-formatted...

10/3,K/20 (Item 20 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01020651 SUPPLIER NUMBER: 00512334
Getting the Message Across.

Bidmead, C.
Practical Computing, v6 No.11, p118-119
Nov., 1983
DOCUMENT TYPE: evaluaton ISSN: 0141-5433 LANGUAGE: ENGLISH
KEYWORD TYPE: ABSTRACT

...ABSTRACT: operating systems have copy utilities for text files but cannot transfer all types of data files. The Dump and Undump utilities of CP-M-80 **convert** files into **ASCII** **code** and back again for transfer but these utilities cannot handle large files or a number of files. The BSTAM software package was developed for telecommunications...

...be used in batch mode without supervision to send a complete disk of files. But BSTAM has to be patched for the individual system with **assembler** routines. Ascom is a sophisticated package that permits a single terminal to drive the communication process. Help routines and menus can be used to direct...

10/3,K/21 (Item 1 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

02302070 Supplier Number: 59122557 (USE FORMAT 7 FOR FULLTEXT)
Whitehill Technologies Launches Whitehill Enterprise(TM) at Softworld Expo.
PR Newswire, p9798
Feb 1, 2000
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 545

... Enterprise was created as a direct response to a need in the professional services industry to take information from multiple legacy data sources, and quickly **assemble** it into professionally-imaged documents that are distributed with ease," she says. "By automating that process, firms can save huge amounts of time and money. Whitehill Enterprise simplifies critical business processes like no other product can."

Whitehill Enterprise **transforms** multiple **data** streams (including **ASCII** , AFP, Metacode & DJDE) into highly readable business documents that are automatically delivered to internal or external recipients. Customized reports as well as complete billing packs...

10/3,K/22 (Item 2 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)

01119684 Supplier Number: 40913553 (USE FORMAT 7 FOR FULLTEXT)
QUAD MOVES TO MEET DEMAND FOR SMT ASSEMBLY AUTOMATION
News Release, p1
August 28, 1989
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 534

... into CIM (computer integrated manufacturing)
strategies -- has introduced a new database software package, AutoProgram (TM) 3.0. An option to the company's QuadStar (R) **Assemblers** , AutoProgram adds "pick and place databases" to reduce **Assembler** programming, setup, and changeover time, thereby reducing the cost of SMT **assembly**
. The software, which runs on a Compaq (R) or IBM(R) PC, also adds capabilities for materials management and hard copy documentation.

Programming time is reduced by generating the **Assembler** 's population routines automatically from databases, instead of "teaching" or

manually programming the **Assembler**
. The databases can be created from
other factory computers, such as a CAD system for component placements and MIS/MRP systems for SMT parts libraries, when the **data** is **converted** to Quad-specified **ASCII** formats.

Fast setup and changeover times result from working with the wealth of database information available. AutoProgram 3.0 provides "feeder changeover instructions" which compare the **Assembler** 's current feeder setup to a new configuration. Feeders that need to be changed or moved to run a new board type are identified quickly on a printed report. Also helping operators with fast changeovers are reports that sort the **Assembler** 's pick and place program by sequence step number, by part number, by pickup number, and by reference designator. Other hard copy printouts are provided...

10/3,K/23 (Item 3 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2004 The Gale Group. All rts. reserv.

01042363 Supplier Number: 40064075 (USE FORMAT 7 FOR FULLTEXT)
AVAILABLE FROM M4 DATA, 9 TRACK TAPE SUBSYSTEM FOR IBM PC/XT/AT USERS
PR Newswire, pN/A
May 27, 1987
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 277

... standard features. The self-loading, self-threading 9800 tape drive uses standard 7" reels. Have your choice of menu driven simplicity, command line syntax, or **assembler** subroutines. Software utilities, included, allow reading a wide variety of file-structured tape data formats, and provide **ASCII** /EBCDIC **code conversion** at the PC for reading mainframe generated tapes and for creating tapes for mainframe use.

Recently formed to target the tape streamer market, M4 Data...

10/3,K/24 (Item 4 from file: 621)
DIALOG(R)File 621:Gale Group New Prod.Annou.(R)
(c) 2004 The Gale Group. All rts. reserv.

01019069 Supplier Number: 39672677 (USE FORMAT 7 FOR FULLTEXT)
80386 Floating Point Library
PR Newswire, pN/A
Jan 10, 1986
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 298

... 12

Subtract	15	20
Multiply	9	16
Divide	10	20
SQRT	15	22

The libraries also include trigonometric functions (sine, cosine, tangent, arctangent); exponentiation functions; **data conversion** procedures (**ASCII**
to/from floating point and Intoger to/from floating
point); and floating point utility procedures. The code space for FPAC is 3200 bytes while the...
...precision library (DPAC), which includes
FPAC, is available for a one time fee of $1250. The packages are delivered in 80386 native (32 bit) source **assembly** for ease of customization and inclusion into user applications. These packages

complement the U S Software line of FPAC/DPAC floating point libraries for the...

10/3,K/25 (Item 1 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)
(c) The Gale Group. All rts. reserv.

01106216 Supplier Number: 41241116
Verdreifachung des Umsatzes geplant
Handelsblatt, p18
March 26, 1990
Language: German; NONENGLISH Record Type: Abstract
Document Type: Magazine/Journal; Trade

ABSTRACT:
...price up from DM998 to DM1,700 to finance the increase in capacity needed. The company employs 25 people and has its products produced and **assembled** by other companies. The card not only enables a PC to be used as a telecopier, but will also **translate** fax messages into **ASCII** **data** form. Profit from this card will be invested in Taiwanese production of the latest product, which integrates PCs with scanners and laser printers into a...

10/3,K/26 (Item 1 from file: 160)
DIALOG(R)File 160:Gale Group PROMT(R)
(c) 1999 The Gale Group. All rts. reserv.

01578160
United States Software Corporation today announced the immediate release of a 68HC11 Floating Point Library.
NEWS RELEASE February 5, 1987 p. 11

... Point Library.This complete software math package operates on the 68HC11. The package is designed in a convenient modular format and is delivered in source **assembly** for maximum user flexibility. The library conforms to the IEEE 754 Floating Point Standard. The 68HC11 FPAC/DPAC can be delivered in either single precision...

... 4.9 KFLOPS), while DPAC operates at 400 FLOPS. The 68HC11 math library also includes trigonometric functions (sine, cosine, tangent, arctangent), logarithmic functions, exponentiation functions, **data** **conversion** procedures (**ASCII** to/from floating point and integer to/from floating point). and floating point utility procedures. A fully implemented FPAC utilizing all operations and functions requires...

10/3,K/27 (Item 1 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

11178623 SUPPLIER NUMBER: 54836547 (USE FORMAT 7 OR 9 FOR FULL TEXT)
National Data.
Survey of Current Business, 79, 5, D-2
May, 1999
ISSN: 0039-6222 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 13194 LINE COUNT: 04572

... Imported
new autos(2) 65.6 58.5 69.0 68.6
(1.) Consists of final sales and change in business inventories of new autos **assembled** in the United States.
(2.) Consists of personal consumption expenditures, producers' durable equipment, and gross government investment.
NOTE.--Chained (1992) dollar series are calculated as...for the chain-type quantity indexes uses weights of more than one period, the corresponding chained-dollar estimates are usually, not additive,

CCAdj Capital consumption **adjustment**
IVA Inventory valuation **adjustment**
(TABULAR **DATA** B.12 NOT REPRODUCIBLE IN **ASCII**)
C. Historical Tables
The Tables in this sectin are derived from the "Summary National Income and Producut Series" tables that were published in the August...

10/3,K/28 (Item 2 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

08656717 SUPPLIER NUMBER: 18254432 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Converting binary to ASCII.
Eisen, Alexander
Electronic Design, v44, n6, p144(2)
March 18, 1996
ISSN: 0013-4872 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 724 LINE COUNT: 00054

... of any nibble will never exceed "9," thus eliminating the necessity for the additional checks.
The program code shown is written Parallax version of PIC **assembler** (for Microchip series RISC 8-bit microprocessors, such as PIC16C5X), and converts 24-bit long binary code, located in three consecutive file registers bin...

...0-3. Each BCD digit then is **converted** into **ASCII** **code** and placed into the xmt...

10/3,K/29 (Item 3 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

08656597 SUPPLIER NUMBER: 18252584 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Image-managing databases: really handling images will remake workflow.(ROI)
Yencharis, Len
Advanced Imaging, v11, n3, p14(3)
March, 1996
ISSN: 1042-0711 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 2170 LINE COUNT: 00179

... as a publishing arena is really a natural place for databases to have immediate impact - affecting where, when, and how the data is to be **assembled** , how it's delivered via server, and to what location is central to this IT industry shift. The database models from Illustra ands Net Objects...

...production workflow models cannot be easily integrated into the newer models of client-server and distributed databases. The relational index of images, text and non- **ASCII** **data** have to be **translated** into a model on which the big relational database providers now have a stranglehold.
The Web, some seem to forget, is supposed to be about...

10/3,K/30 (Item 4 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

06511610 SUPPLIER NUMBER: 14375403 (USE FORMAT 7 OR 9 FOR FULL TEXT)
C compilers for 8-bit microcontrollers. (includes related articles on future microcontroller designs, the C programming language and on programming microcontrollers) (Buyers' Guide) (Special Report) (Cover Story)
Shear, David
EDN, v38, n13, p116(9)
June 24, 1993

DOCUMENT TYPE: Cover Story ISSN: 0012-7515 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3766 LINE COUNT: 00278

...ABSTRACT: for an accurate evaluation of performance. Optimization can be achieved through various techniques, each of which are discussed. Other factors that affect efficiency include the **op - code** library's quality, **data** size **conversion** , type of variables used and type of memory mapping scheme. Two options that are potentially beneficial are in-line **assembling** and floating-point operations. American National Standards Institute C is an alternative with advantages of its own.

10/3,K/31 **(Item 5 from file: 148)**
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

04123572 SUPPLIER NUMBER: 08009665 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Tracing in a digital age. (automated tracer mill)
Beard, Tom
Modern Machine Shop, v62, n5, p54(9)
Oct, 1989
ISSN: 0026-8003 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 2423 LINE COUNT: 00187

... application demonstrates how it all works. The digital data file is first brought into a personal computer running Sytos software (Sytron Corp., Westborough, Massachusetts) which **converts** the binary **code** into an **ASCII** file. **ASCII** is a standardized format that makes it possible to exchange data files from one computer to another.

In this case, the file is then transmitted...

...the massive amount of data in the file, it may take ten minutes just to be loaded.

Once in CAD, the data points are first **assembled** into a series of parallel curves, 0.030-inch apart in model size, which essentially correspond to the path of the tracer probe. These curves...

10/3,K/32 **(Item 6 from file: 148)**
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

03928755 SUPPLIER NUMBER: 07774111 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Breaking down barriers to low-cost manufacturing.
Coreson, David E.
Machine Design, v61, n13, p83(5)
June 22, 1989
ISSN: 0024-9114 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 2839 LINE COUNT: 00238

... days with the vendor's engineers, providing training and process engineering consultation to resolve the problems. Data integration

Integrating design data into manufacturing planning and **assembly** is important to low-cost manufacturing. At Tektronix, engineering, manufacturing, and the etched circuit board facility, 30 miles away, each had a separate database.

Custom software called TCADPCB was developed to consolidate the three databases. It **converts** CAD **data** to **ASCII** , lets **operators** graphically specify IC insertion paths, and produces code to drive insertion equipment. The software reduced prototyping time, errors, and tolerances, improving quality and allowing quick...met all goals, is priced nearly 50% lower, and has a final assembly and test time of only 10 min. In addition, overall manufacturing and **assembly** labor was trimmed by more than two thirds. The product costs 35% less to manufacture, and its MTBF tripled. ONE SYSTEM INSTEAD OF THREE

Custom...

...to four weeks.

TCAD-PCB software, based on the company's TekniCAD drafting package, performs three functions. First, using a combination of IGES and custom **translation** routines, it **converts** **data** from different CAE databases to **ASCII** . It then gives the operator a graphical display of the circuit board, which the operator interactively edits to specify the IC insertion path. Finally, TCAD...

10/3,K/33 (Item 1 from file: 624)
DIALOG(R)File 624:McGraw-Hill Publications

0465720
No Business Like Show Business: Presenting business and presentation graphics packages for UNIX
Unix World March, 1993; Pg 94; Vol. X, No. 3
Journal Code: UNIX ISSN: 0739-5922
Section Heading: PRODUCT REVIEW
Word Count: 2,273 *Full text available in Formats 5, 7 and 9*

BYLINE:
Natalie Engler

TEXT:
...caption style, and line and fill color. You can open any palette and pin it to the window for quick access.

When it comes to **assembling** your presentation, Ta-Dah is less versatile than the other programs. The only way to apply a consistent background is to create a slide, then...

... increased datasets, and an increased number of data rows. For an extra $200 Dux will bundle another program, X-plot, with Ta-Dah. X-plot **converts** **ASCII** **data** to XY plots.

Although it lacks an outliner, Ta-Dah lets you add slide notes, which you can use as handouts. You can print slides...

10/3,K/34 (Item 2 from file: 624)
DIALOG(R)File 624:McGraw-Hill Publications

0368405
Modern Editors for UNIX
Unix World February, 1992; Pg 73; Vol. IX, No. 2
Journal Code: UNIX ISSN: 0739-5922
Section Heading: Product Review
Word Count: 3,259 *Full text available in Formats 5, 7 and 9*

BYLINE:
Rick Farris

TEXT:
... available in the MS-DOS version of Vedit, and is planned for a future revision of the UNIX version. Surprisingly, Vedit includes some help for **assembly** language programming, because it knows the difference between an **assembly** language comment and the **opcode** and operand fields, and **converts** the **code** to uppercase letters, while leaving the comments in mixed-case.

Vedit provides basic word-processing support, with better indentation than most text editors. Like Edix...
... the time and inclination, it's great for creating off-the-cuff macros. The command #s/w;y>// will strip all the comments from an **assembly** language file, for instance.

Vedit isn't as full-featured as editors like Slick, but it is blindingly fast, particularly in its MS-DOS version...

10/3,K/35 (Item 1 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

01103956 97-53350
A new manufacturing format
Andreiev, Nikita
Printed Circuit Design v12n10 PP: 11-15 Oct 1995
ISSN: 1047-5567 JRNL CODE: PCC
WORD COUNT: 3168

...TEXT: retained the good features and slightly modified them for modern use.

Other Sections

By the same token, the other sections can be BARE BOARD TEST, **ASSEMBLY** , IN-CIRCUIT TEST (ICT), BILL OF MATERIALS (BOM) or whatever is human and machine readable. The reason the code is in ASCII rather than a...

...example, many AD systems put out a code that is almost standardized. Why not use it? Or take the BOM. Many CAD systems generate this **code** in **ASCII** , which with minor **adjustments** , can become a standard and, therefore, usable in its present form.
Even test programs can be simplified to a common denominator. The same can be...

10/3,K/36 (Item 1 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext

00631637 CMP ACCESSION NUMBER: EET19890116S4437
User groups fine-tune Microsoft C tools
RICHARD DOHERTY
ELECTRONIC ENGINEERING TIMES, 1989, n 521, 17
PUBLICATION DATE: 890116
JOURNAL CODE: EET LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: 521PG17
WORD COUNT: 510

... Microsoft said that results in speed improvements of between three and four times that of earlier Quick C releases.

A new feature, an in-line **assembler** , allows users to edit, compile/ **assemble** and debug C (and in-line **assembly** code) without leaving the environment. Microsoft claimed it is the first package to include this real-time companion **assembly** feature, which normally requires the purchase of a separate **assembly** package.

An over-the-shoulder reference system, dubbed QC Adviser, acts as a programming coach. Through it, the programmer has access to a large on-line reference library of C protocols and definitions.

Other reference works in the QC Adviser include **ASCII** **conversion** charts, printer reference **code** lists, escape sequences and other tabular reference works. Like Microsoft's far more extensive CD-ROM- based Programmer's Library, the package offers several examples...

10/3,K/37 (Item 1 from file: 674)
DIALOG(R)File 674:Computer News Fulltext

02364
IBM extends packet-switched capability

Byline: Elisabeth Horwitt, CW Staff
Journal: Computerworld Page Number: 65
Publication Date: November 20, 1989
Word Count: 409 Line Count: 29

Text:
... and non-SNA terminals onto a single Synchronous Data Link Control backbone network. The second release can also act as an X.25 and packet **assembler** /disassembler that provides protocol and **data** stream **conversions** for **ASCII** terminals, allowing users to access both SNA and X.25 connections, IBM said.

A programmable interface allows users to provide support for whatever types of...

10/3,K/38 (Item 2 from file: 674)
DIALOG(R)File 674:Computer News Fulltext
(c) 2004 IDG Communications. All rts. reserv.

002242
Future version will integrate disparate data on X.25 links.
Byline: Tom Smith, New Products Editor HDNew IBM software lets users blend asynch ASCII traffic with SDLC
Journal: Network World Page Number: 6
Publication Date: November 13, 1989
Word Count: 568 Line Count: 41

Text:
...25 and ASCII data on X.25 links.

The software, Programmable Network Access (PNA) Version 1.0, runs on an IBM Personal System/2 and **converts** asynchronous **ASCII** **data** into a synchronous EBCDIC data stream for transmission over a single line. The software will be available in June 1990.

PNA Version 1.1, the availability of which will be announced in April 1990, comes bundled with a packet **assembler** / disassembler for transporting SNA, native X.25 and ASCII data over a single X.25 connection.

IBM will also offer both products in C-based...

10/3,K/39 (Item 1 from file: 810)
DIALOG(R)File 810:Business Wire
(c) 1999 Business Wire . All rts. reserv.

0151374 BW692

IBM: New IBM software allows customers to consolidate data networks, protect equipment investments

November 7, 1989

Byline: Business Editors And Computer Writers

...terminals onto a single Synchronous Data Link Control (SDLC) backbone network.

With the second release, it can serve as an X.25 concentrator and packet **assembler** /disassembler (PAD) and offers protocol and **data** stream **conversions** for **ASCII** terminals, allowing users of those terminals to access SNA or X.25 applications more easily.

14/9/11 (Item 1 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)

06136660 Supplier Number: 53892788 (THIS IS THE FULLTEXT)
Tool Suite Enables Designers To Craft Customized Embedded Processors.
Bursky, Dave
Electronic Design, v47, n3, p33(1)
Feb 8, 1999
ISSN: 0013-4872
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 2629
TEXT:
When Standard CPU Cores Don't Fit The Bill, A Malleable Core And Tool Suite Allows You To Create An Optimized Solution.

Catalogs from many chip. and intellectual-property (IP) suppliers are filled with dozens of standard 8-to 64bit CPUs, as well as cores for embedded applications. Unfortunately, these items often carry too much "baggage" or can't be optimized for the desired application. A few companies already address such issues by allowing designers to strip out or add some features, change cache sizes, or add instructions. But even that can fall short of delivering an optimum solution for embedded applications.

A more comprehensive solution has been developed by Tensilica Inc., Santa Clara, Calif. Designers can optimally tune the CPU architecture, instruction-word size, and other features to the targeted application by using a very logic-efficient CPU and an **instruction - set architecture** , dubbed Xtensa. A suite of Tensilica design tools makes it a simple task to rapidly generate a system-on-a-chip with an optimized processor and instruction set. Complementing the design tools is a suite of program development tools including an ANSI C/C++ compiler, an **assembler** , a debugger, a linker; a **code** profiler, and an instruction-set simulator built on top of the popular GNU tools. These tools also are extensible. As **new instructions** are **added** , they're automatically enhanced with the **new instructions** , as well.

Since each application's requirements are unique, configurability is desirable and beneficial in embedded-processor design. Applications can be differentiated by examining program size, the instruction mix, and the frequency and degree of instruction and data access. Some factors that may affect system performance are the interrupt response time, context-switch overhead, and I/O performance.

The sheer number-crunching performance required of the processor is a key factor in embedded applications, such as image compression, speech coding, and protocol conversion. This may require the CPU to include specialized processing blocks or optimized instructions to more efficiently execute key algorithms. Plus, many task-oriented embedded applications are interrupt-driven and must deliver predictable worst-case performance. Frequent context switches are common to such applications. A large percentage of these switches typically are initiated by interrupts. So, fast context-switching time is critical. The embedded controller also must he able to service interrupts quickly.

Some applications also lend themselves to having a particular on-board coprocessor tightly integrated on-chip. Disk-drive controllers execute algorithms based on repetitive state-space equations. In this case, the addition of special-purpose hardware, such as an onboard DSP coprocessor with a dedicated multiplier-accumulator (MAC), would be a more optimized design solution to speed processing.

In other applications, the degree of instruction and data locality is highly variable. A network protocol converter that strips off a portion of a TCP/IP header and looks at the Nth bit of data always has new data flowing through the machine. A large data cache, then, is of little benefit. But bit-manipulation instructions are invaluable. For robotics applications, the instruction access rate must be very high due to the real-time control requirements. A large instruction cache in these applications can waste time and money. By contrast, some laser-printer controller and telecommunications switching applications have a high instruction-cache locality. Having the right amount of instruction cache can be very effective.

Quick Customization

To customize their processors to address these factors, designers can turn to the Xtensa architecture and design system. At the heart of this system are the software tools that configure both the processor logic and instruction set. They not only permit system designers to configure and extend the 32-bit synthesizable CPU core, but also provide an automated way of adding user-defined, application-specific instructions. These extensions to the instruction-set architecture can replace time-consuming, repetitive inner loops of application code. For example, such customized commands can be used where the application requires a particular combination of subtractions, compares, and absolute values and additions, or where the algorithm is always shifting and extracting a particular collection of bits. This can help accelerate execution speed by as much as a factor often.

The 25,000-gate, 0.25-[micro]m CMOS Xtensa processor consists of a core that can execute a 16/24-bit instruction set. This core occupies 1 to 1.5 [mm.sup.2] of area and delivers an integer throughput of up to 300 MIPS (based on the Dhrystone V2.1 benchmark and a 250-MHz clock). When powered by a 2.5-V supply, the core dissipates about 0.5 mW/MHz. So at 250 MHz, power consumption peaks at 125mW

The basic core consists of the 32-bit data path with a five-stage pipeline and the ability to handle 32-bit addressing and data (Fig. 1). A memory-protection unit, branch-logic and instruction-fetch control, and a host-system interface also are part of the core.

Tensilica's engineers paid considerable attention to minimizing power consumption. The architecture includes features that implement dynamic power-down of unused circuit blocks and a software wait instruction. The RTL design file can readily be ported to low-voltage cell libraries when the CPU is synthesized. An interactive, on-line chip estimator (target speed, gate count, and power) guides the processor definition process so the designer can see how each change affects chip size, power, and performance.

A rich instruction set of over 70 commands, including powerful bit-manipulation and jump/compare instructions, is incorporated in the core. The base instruction set was designed for high code density. It includes both 16- and 24-bit versions of the commands, rather than the more commonly used 32-bit encoding. The set also uses a register-window scheme as part of the programming model (similar in some ways, hut different in others, to the register-window approach used in Sun's SPARC processor family). The register-window approach provides the processor with fast context switching, since only the reference pointers must be switched when the context has to change.

Powerful branch operations (combined compare and branch, no delay slot, and zero-overhead looping) and better bit manipulation (funnel-shift, bit-test-and-branch, and field-extract operations) are included in the instruction set. The core also supports speculation and instruction-level parallelism (conditional moves, speculative loads, and multiple operations per instruction). Compared to other popular RISC cores-such as the ARM Thumb or the MIPS II--software run on the Xtensa core is about 10% smaller, does not require any mode switching, and incurs no performance penalty.

The Xtensa processor generator (XPG), a software tool with a browser-like graphical user interface, complements the core (Fig. 2). The software enables the user to easily add additional functionality to the processor core, in terms of both hardware options and instruction-set extensions. And by simply entering the desired performance level, power dissipation, and area goals, the designer can affect attributes that are dependent upon the silicon process used.

Complete Flexibility

The configuration editor in the XPG software helps the user define the processor hardware options. It specifies all processor interfaces, such as its memory and cache subsystem. This includes setting up the memory organization (address and data-bus width). Initially set at 32 bits, the data-bus size can optionally be set for 64 or 128 bits. Also definable are the size of the register file (32 or 64 32-bit registers), the instruction and data caches (1,2,4,8, or 16 kbytes), the cache line (16, 32, or 64 bytes), the write buffer (4 to 32 entries), and the register window (32 or 64 entry registers). The generator also features a simple memory-protection unit that controls access rights and cache characteristics for each of

eight segments in the address space.

Coprocessing hardware functions, such as a 16-bit multiplier, a 16-bit MAC, register file size, and cache sizes, can be added merely by clicking a mouse. Plus, designers can add cache test instructions and a write buffer (4/8/16/32 entries). They also can select big-endian or little-endian byte ordering. Other user-selectable features at the click of a mouse include kernel/user modes, the number of external interrupts (0 to 32), the number of interrupt priority levels, multiple 32-bit timers, address-trace capabilities, and instruction/data breakpoint support. Future options will include integer 32-bit multiply/divide capability, as well as floating-point math support.

The XPG software's result is a semiconductor-manufacturer-independent synthesizable RTL file. This can be turned into a gate-level file by selecting a manufacturer's cell library and synthesizing the design. Optimized design scripts are available for use with leading synthesis tools, such as Ambit's BuildGates and the Synopsys Design-Compiler. The design flow supports most ASIC standard-cell tool flows, providing a familiar environment and giving designers the leverage of selecting from a large number ASIC suppliers (Fig. 3).

Once a processor configuration is generated, the software execution's real-time characteristics can be analyzed by using the instruction-set simulator (ISS) provided by Tensilica. A series of architectural changes and iterations then can be performed to further optimize the processor subsystem's overall performance. After a suitable configuration is identified, the processor is generated along with all of the tools and support scripts needed to complete the software development and debugging effort. At the same time, the ASIC design and chip implementation can be finished.

To extend the instruction set, designers fire up the Tensilica instruction-extension (TIE) language. This language accepts as its input a Verilog-like syntax that describes the custom function. The output after compilation is a correct-by-construction HDL. It also contains patches for the compiler, assembler, simulator, and debugger that can automatically integrate into the tool suite.

Part of the instruction-set customization is the definition of the exception layer. The designer can build a prioritizable interrupt structure upon this exception layer. Thus, instead of having to settle for a fixed interrupt structure and the context-switch characteristics of a given processor core, the Xtensa architecture lets the designer support as many combinations and levels of interrupts as needed.

The exception layer is optional, though. If the design is not interrupt-driven, the Xtensa generator omits the unnecessary hardware. If the application involves numerous interrupts, multitasking, or the need for very fast context switches, the Xtensa core supports these needs as well. For designs with timers/counters, the interrupt-function unit should be enabled in most cases. Users can do this merely by checking a box on the XPG form.

The Xtensa interrupt-configuration options help the designer match system requirements and hit real-time performance targets. For instance, the configuration editor lets the user select different options for the interrupt structure. The processor employs a three-level, priority-encoded interrupt scheme, giving the user a choice of high-, medium-, or low-priority interrupts. Level 1 is used for exceptions and low interrupts. Medium- and high-priority interrupts operate at Levels 2 and 3, respectively.

The Xtensa processor's unique, variable windowed-register architecture depends on the use of the exception layer in the processor. Its overlapping window registers are designed to speed procedure switching time. Furthermore, the ability to set the number of registers in each window lets the designer define how many simultaneous procedure switches the system can support. The overlapping registers between windows primarily are used for parameter passing. Their use increases program performance by eliminating the register saves and restores at procedure entry and exit, as well as argument shuffling during call operation. It allows more local variables to exist permanently in registers.

The variable windowed-register option can support either 32 or 64 physical registers. Implementation of the most cost-effective level of procedure call performance is allowed by the choice of 32 or 64 physical

registers. Variable windowing enables register use at peak efficiency, regardless of the number of implemented registers. The largest register window has 16 visible registers at any one time. And the offset for where the register window is looking is variable, rather than fixed. This means the compiler automatically can choose the optimal shift of the register window on each procedure call and return. When there are 32 registers, the design supports windowing and procedure calls up to five levels before any registers are automatically spilled to memory. With 64 registers, the designer can have up to 12 procedure levels without hardware-controlled spilling.

Xtensa's variable windowed-register application binary interface (ABI) provides an efficient call/return process in terms of execution time and code size. There are separate stack- and frame-pointer registers to handle cases where they would differ by a constant. The variable windowed registers are designed to handle four key scenarios:

1. Support for per-call, variable windowed-register increments
2. Support for the use of a single stack for both register save/restore and local variables
3. Support for variable frame sizes
4. Support for programming-language exception handling

The designer has a very flexible and high-performance fast context-switching resource that can be used for many applications demanding highly responsive controls.

Keeping The Bugs Out

One other key feature available through the XPG tool is the "Enable Debugging" box, which can be checked during configuration. When selected, this box causes additional hardware to be generated as part of the processor. Then, it implements the on-board debug module. The debugging port is used to debug the processor in the context of its interactions with other peripherals on the system-level chip. It provides direct access and visibility to the core Xtensa processor embedded in the system-on-a-chip ASIC design.

The on-chip debug capability permits the designer to enable breakpoints on data or address points, stopping the processor. After that, all internal registers and counters can be read out. An industry-standard JTAG interface can be used to serially clock in and out the address and data locations that the designer wants to read. Another debug option, a simple trace port, provides a complete real-time trace of key pipeline state, control, and address information. With an on-chip logic analyzer or an off-chip emulator pod, it can capture the status of important buffers after the break. Typically, the trace buffer exists in a hardware emulator pod.

The XT1000 emulation card, which is based on a large FPGA, also is available in the suite. It can be employed to emulate the CPU or full digital system simply by synthesizing the Xtensa core and other logic/instruction extensions into the FPGA's cell library. In a few hours, users can evaluate a fully functional design from final processor definition.

To support the development of hardware and software, Tensilica has formed partnerships with various tool and silicon suppliers. Keeping emulator cards in mind, the company has worked with Altera to ensure that the core can be ported to the FLEX family of FP GAs. For design tools, alliances have been struck with Arcadia, Artisan, Avanti, Cadence, Synopsys, Virtual Silicon, and Virage Logic. Likewise, Tensilica is working with Wind-River Systems to port the VxWorks real-time operating system to the Xtensa core.

The company plans to offer its Xtensa technology in two forms. First, there will be a soft IP solution (a collection of RTL-based design files, scripts, and tools) that designers can utilize to roll their own physical design. Second, users can obtain a hard IP block plus tools (users get a physical implementation of the core with the desired features), with the processor supplied as a physical layout in the customer-specified cell library guaranteed for a performance level.

PRICE AND AVAILABILITY

The Xtensa core and its supporting software is available in two forms. The standard option comprises a Verilog or VHDL RTL description, various EDA tool script files, a test suite, placement guidelines, and the software tool chain. The second is a more "rugged" solution that comes with

a Verilog or VHDL netlist, a GDS II hard layout of the core using the cell library specified by the customer; a test suite, and the software tool chain. For the standard, the core will cost $250,000 for a single-instance manufacturing license, plus a negotiated royalty per core shipped. Pricing for the second option is negotiated on a case-by-case basis. An FPGA-based emulation board, the XT1000, sells for about $10,000. The board allows designers to "download" the core after customization to evaluate functionality in just hours.

14/3,K/1 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02757642 SUPPLIER NUMBER: 110734571 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Motorola enhances StarCore DSP: SC140e core offers new instructions, caches, and task protection.

Halfhill, Tom R.
Microprocessor Report, 17, 10, 20(6)
Oct, 2003
ISSN: 0899-9341 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 4332 LINE COUNT: 00403

... the SC140e allows user-level tasks to see and modify only a copy of the status register. This also preserves compatibility with existing SC140 software.

New Instructions Accelerate Multimedia

The StarCore SC1000 instruction-set architecture (ISA) is already well suited for data-intensive communications and multimedia processing. It has numerous instructions...

...Viterbi encoding. (For a large table of the most relevant instructions, see the May 1999 MPR article referenced earlier.)

The SC140e adds about a dozen **new instructions** to the existing ISA, including some instructions that handle smaller data types in 32-bit chunks. For example, one instruction loads four eight-bit operands into four different registers in parallel--ideal for byte-sized multimedia data. However, Motorola is not publicly releasing information about all the **new instructions** at this time. The **instructions** will probably be disclosed next year, when StarCore LLC absorbs them into the future SC2000 ISA.

Like existing SC1000-based DSP cores, the SC140e supports the instruction-set accelerator plug-in, a coprocessor interface for adding application-specific instructions and registers. It allows Motorola to extend the standard **ISA** for its own chips or for SoCs designed to customer specifications by Motorola. Extensions can be written in Verilog or VHDL and integrated with the...

...instructions behave like standard instructions--including the ability to run in parallel with standard instructions in the same VLIW bundles. Software programmers can use extension **instructions** in **assembly** language, or by defining intrinsic **functions** with the Metrowerks CodeWarrior ANSI C compiler or a third-party compiler. (The Green Hills MULTI tools support StarCore DSPs with a C/C++ compiler...

...not in assembly language. It's a statically scheduled VLIW DSP with variable-length instructions, variable-length instruct/on bundles, 16 function units, and an **ISA** with nearly 200 instructions, not counting the numerous addressing variations and optional instruction prefixes. Because of the architecture's complexity, Motorola expects programmers to reserve assembly language for tight inner loops and critical routines, relying on C for the bulk of the code. Some **new instructions** in the SC140e are intended to improve the performance of **compiled code** --like the four-byte parallel load, which a compiler can substitute for four single-byte loads.

Competition From Analog Devices and Intel

Although MMUs, address...

14/3,K/2 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02744833 SUPPLIER NUMBER: 108783678 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Tuned up: 30th annual microprocessor directory: new processor offerings continue to deliver optimized configurations that provide "tuned" processing engines for specific applications.

Cravotta, Robert
EDN, 48, 19, 45(11)
Sept 4, 2003
ISSN: 0012-7515 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 7506 LINE COUNT: 00658

14/3,K/3 (Item 3 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

02694730 SUPPLIER NUMBER: 99131117 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Soft cores gain ground: key trends are higher speeds, better architectures, configurability.
Halfhill, Tom R.
Microprocessor Report, 17, 2, 28(7)
Feb, 2003
ISSN: 0899-9341 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 5695 LINE COUNT: 00512

... was FLIX (flexible-length instruction extensions), which adapts VLIW concepts for communication, multimedia, and networking applications (MPR 11/25/02-06, "FLIX: The New Xtensa **ISA** Mix"). And in November, Tensilica won three new U.S. patents for its configurable-CPU technology (MPR 12/9/02-01, "Tensilica Patents Raise Eyebrows...the essential advantages of a 32-bit RISC architecture. For one thing, there's no need for explicit mode-switching between 16-and 32-bit **instructions** . **Assembly** ...language programmers and **compilers** can freely mix both types of instructions in the same code without restrictions. ARCtangent-A5 automatically decodes the different instructions before feeding them into a ...for extra instructions to shuffle data between registers.

In addition to code compression, the ARCompact ISA offers other improvements. It has several new 32-bit **instructions** , such as a compare-and-branch instruction that does the work of two previous instructions and some bit-manipulation operations of particular value for networking applications. The ARCompact **ISA** also greatly expands the number of opcode slots available for user-defined custom instructions. Developers can now add as many as 128 16-bit instructions and 128 32-bit instructions, compared with 69 user-defined instructions in the A4 **ISA** . As before, any of these instructions can support conditional execution based on 16 predefined or 16 user-defined condition codes.

One disadvantage of ARCompact is some additional complexity in the processor's decoder logic, which must scan each instruction header to determine the **instruction** length. The **additional** gate delays in this critical path ...compared with that of the A4. The difference is relatively small--perhaps 10%--and is partly or wholly offset by the improved efficiency of the **new** **ISA** and the **instruction** cache. (Shorter instructions effectively increase the size of the cache, so fewer cycles are wasted on cache misses.)

Another drawback of ARCompact is binary incompatibility with **code compiled** for earlier ARC processors, but migrating to the new **ISA** isn't difficult. The programmer's models for the two instruction sets are similar. In many cases, developers ...addition, a new switch in ARC's MetaWare High C/C++ compiler automatically generates binary executables that substitute 16-bit instructions for many 32-bit **instructions** , so hand-coded in-line **assembly** language isn't required.

14/3,K/4 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

02380994 SUPPLIER NUMBER: 60103147 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Next Gen Embedded Firmware Speeds Up Java 6-Fold 03/14/00.(Company Business and Marketing)
Dennis, Sylvia
Newsbytes, NA
March 14, 2000

LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 599 LINE COUNT: 00055

... like the C programming language, while the program code is multi-platform, the downside is that the code's execution speed is generally slower than **assembler code** for a single computing platform.

In a desktop computer environment, this is not a major problem, but on a portable or embedded firmware system the...

...more resource-constrained embedded devices.

Hardware approaches, meanwhile, involve the use of Java-specific microprocessors and require significant hardware and software investments associated with any **new** CPU (central processing unit) **instruction** set.

Jedi says its approach is more adaptive, since JStar enhances the processor that is already being used, giving it the ability to run Java...

...more quickly than software interpretation.

In addition, the firm says, JStar operates directly on Java byte [illegible], which eliminates the extra memory required by JIT **compilers** to [illegible] native **code** they generate.

Perhaps most importantly of all, adding JStar requires no changes to the microprocessor's instruction set or pipeline architecture.

This means that operating...

...long instruction word) and multi-issue processors.

Jedi's first implementation of the JStar accelerator technology is on an R3000 class processor supporting the MIPS **instruction set architecture** and Sun's PersonalJava 3.0 technology.

Jedi's Web site is at http://www.jeditech.com .

Reported by Newsbytes.com, http://www.newsbytes.com...

14/3,K/5 (Item 5 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01288597 SUPPLIER NUMBER: 07060390 (USE FORMAT 7 OR 9 FOR FULL TEXT)
The Clipper (TM) Processor: instruction set architectures and implementation. (product announcement) (technical)
Hollingsworth, Walter; Sachs, Howard; Smith, Alan Jay
Communications of the ACM, v32, n2, p200(20)
Feb, 1989
DOCUMENT TYPE: technical ISSN: 0001-0782 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 9556 LINE COUNT: 00758

... or to provide a reasonable range of features. There were also commercial barriers to using existing architecture. The decision was thus made to design a **new** instruction set architecture, using the previous experience of the designers and the latest thinking in the computer architecture research community.

Fashions in computer architecture have...

...such very simple machines. Some discussion of the RISC/CISC issues appear in [7] and [14].

The choice was thus made to design a new **instruction set architecture** (**ISA**). The **instructions** , the module design, and the functional partitioning were chosen to permit mainframe level performance, and to permit future compatible mainframe implementations. The continuing and increasing adoption of the easily ported UNIX as the standard operating system for academic, software development, and workstation environments made a new **ISA** commercially feasible.

Outline and Context

It is possible to describe a "computer" at many levels. The **instruction set architecture** (ISAe refers to the computer instruction set as expressed in binary or in **assembly** language and its **functions** ; the **ISA** is usually described in the "principles of operation" manual. We use the term design architecture to refer to the highest level description of an implementation, i.e., the block diagram and parameter level. Below

that are gate and circuit level descriptions.

This article focuses primarily on CLIPPER's **instruction set architecture** , and examines the design architecture and related issues such as performance, design tradeoffs, design implications, and areas for possible future expansion.

MEMORY ARCHITECTURE AND DATA...

14/3,K/6 (Item 6 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01170855 SUPPLIER NUMBER: 00664929
Floating-point methods combine to boost performance.
Porter, Kent; Kath, James
Computer Design, v25, n3, p75-78
Feb. 1, 1986
ISSN: 0010-4566 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

ABSTRACT: Two new methods of performing floating-point arithmetic have been developed **recently** : the MIS-STD-1750A **instruction - set architecture** and the IEEE Standard 754. The 1750A standard defines sixteen general-purpose and six special-purpose 16-bit registers, two timers, two formats for floating-point representations and a complete **assembly** language that includes floating-point **instructions** , specifying a number of functions, such as memory management, I-O, interrupt vectoring and error recovery. The IEEE standard is more complex and has a...

14/3,K/7 (Item 1 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

05615562 Supplier Number: 106863717 (USE FORMAT 7 FOR FULLTEXT)
ARM grows more thumbs: ARM announces v6 and 32-bit thumb extensions at EPF.
Levy, Markus
Microprocessor Report, v17, n6, p18(4)
June, 2003
Language: English Record Type: Fulltext
Document Type: Newsletter; Trade
Word Count: 2785

... environments; and more than 60 single-instruction, multiple-data (SIMD) instructions. (See MPR 1/2/01-03, "ARM Embraces SIMD Support")

A Sore Thumb

The **new** extensions include 32-bit ARM instructions for improved data handling, but even more interesting is the birth of a new version of Thumb that supports...of both worlds--optimal performance and highest-density code--ARM has created a new Thumb ISA, called Thumb-2 (T2). T2 combines the preexisting Thumb **instructions** with a handful of **new** 16-bit **instructions** and an armful of 32-bit instructions. During development, these 32-bit instructions were known within ARM as the Wrist extensions.

Among the **new** 16-bit Thumb **instructions** is a no-operation instruction (NOP). Besides its "do nothing" functionality, the NOP is useful for ...to code size, part of the reason this instruction was not previously available in Thumb. It is now up to the programmer to choose between **extra instruction** words or higher-performing code.

For improved code density and performance, ARM **added** a "Compare zero and branch" **instruction** to the 16-bit T2 ISA. Taking the form of CZB<condition> ...NE} <label>

Programmers will find this **new instruction** useful in "if" clauses, where tests for pointers with zero values are carried out. It might seem obvious that for increased performance, ARM should have **added** the CZB **instruction** to the ARM ISA mode as well as to T2; however, cores based on the v6 ISA and the ARM1026EJ-S support branch folding, which...

...has always had the luxury of supporting conditional execution of its

instructions. Unlike the ARM ISA, Thumb supports conditional execution only on its conditional branch **instructions** , being unable to afford the **extra** four condition bits on every instruction. Even with the inclusion of 32-bit T2 instructions to help minimize the encoding space, the architecture does not 98% of the conditional blocks generated by a compiler. In highly optimized **assembler** coding, If-Then may not be sufficient.

The If-Then instruction takes the form:
IT{x{y{z}}} <condition>
where x, y, and z indicate...

14/3,K/8 (Item 2 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

05548956 Supplier Number: 101174044 (USE FORMAT 7 FOR FULLTEXT)
MIPS embraces configurable technology: pro series processors with corextend compete with ARC and Tensilica.
Halfhill, Tom R.
Microprocessor Report, v17, n3, p7(9)
March, 2003
Language: English Record Type: Fulltext
Document Type: Newsletter; Trade
Word Count: 6637

... either be oblivious to the instructions or balk at what appeared to be undefined opcodes. Designers must therefore write macros and intrinsic functions that allow **assemblers** and C/C++ compilers to call the instructions, plus dynamically linked libraries (DLL) to describe the operation of the instructions to software simulators. MIPS provides...do with CorExtend. The example demonstrates how a single custom instruction could accelerate a common algorithm in a voice-over-Internet-Protocol (VoIP) application.

The **new instruction** , udi - madd (user-defined **instruction** /multiply-add), performs a 16 x 16-bit multiply-accumulate (MAC) operation. It stores the extended-precision 40-bit result in a user-defined local... ...part of the custom logic) scales the results. MIPS estimates this function block would require 15,000 to 25,000 gates, depending on the synthesis **compiler** . (Without the **function** block, the minimum configuration of an M4K Pro Series core is about 32, ...core with the custom function block is similar to that of an ARCtangentA5 with MAC extensions (about 55,000 gates).

Figure 2 shows the MIPS **assembly** -language **code** for the VoIP algorithm's inner loop, using standard MIPS instructions. The loop consists of 18 instructions (including a NOP in ...slot) that execute in 10-20 clock cycles, depending on the outcome of three conditional branches controlled by input variables. To the right of the **assembly** -language **code** , Figure 2 shows how the whole loop collapses into a few lines of C code by calling a predefined intrinsic function, udi .madd. That function ...has announced 64-bit VLIW extensions that a customer (Conexant) is using for similar applications. (See MPR 11/25/02-06, "FLIX: The New Xtensa **ISA** Mix.")

What MIPS Left Out

For now, the most significant new feature of CorExtend is the ability to add custom instructions without a MIPS architectural...

14/3,K/9 (Item 3 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

04115075 Supplier Number: 54079727 (USE FORMAT 7 FOR FULLTEXT)
MIPS: Comprehensive guidebook to the MIPS architecture now available.
M2 Presswire, pNA
March 10, 1999
Language: English Record Type: Fulltext
Document Type: Newswire; Trade

Word Count: 627

... true expert can deliver. It provides an in-depth, easy- to-use guide to the MIPS instruction set, including special attention to processor control and **assembler** mnemonics for every **instruction** . This **new** volume covers everything from MIPS I to MIPS V **ISA** (**instruction set architecture**) with appendices devoted to the optional MIPS16 and MDMX ASEs (application specific extensions). The MIPS architecture has proven to be a widely adopted and successful...

14/3,K/10 (Item 4 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

04108977 Supplier Number: 54013916 (USE FORMAT 7 FOR FULLTEXT)
MIPS TECHNOLOGIES: Comprehensive guidebook to the MIPS architecture now available.
M2 Presswire, pNA
March 4, 1999
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 519

... true expert can deliver. It provides an in-depth, easy-to-use guide to the MIPS instruction set, including special attention to processor control and **assembler** mnemonics for every **instruction** . This **new** volume covers everything from MIPS I to MIPS V **ISA** (**instruction set architecture**) with appendices devoted to the optional MIPS16 and MDMX ASEs (application specific extensions). The MIPS architecture has proven to be a widely adopted and successful...

14/3,K/11 (Item 1 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)
(c) 2004 The Gale Group. All rts. reserv.

06136660 Supplier Number: 53892788 (USE FORMAT 7 FOR FULLTEXT)
Tool Suite Enables Designers To Craft Customized Embedded Processors.
Bursky, Dave
Electronic Design, v47, n3, p33(1)
Feb 8, 1999
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 2629

... can optimally tune the CPU architecture, instruction-word size, and other features to the targeted application by using a very logic-efficient CPU and an **instruction - set architecture** , dubbed Xtensa. A suite of Tensilica design tools makes it a simple task to rapidly generate a system-on-a-chip with an optimized processor and instruction set. Complementing the design tools is a suite of program development tools including an ANSI C/C++ compiler, an **assembler** , a debugger, a linker; a **code** profiler, and an instruction-set simulator built on top of the popular GNU tools. These tools also are extensible. As **new instructions** are **added** , they're automatically enhanced with the **new instructions** , as well.

Since each application's requirements are unique, configurability is desirable and beneficial in embedded-processor design. Applications can be differentiated by examining program...

14/3,K/12 (Item 2 from file: 16)
DIALOG(R)File 16:Gale Group PROMT(R)
(c) 2004 The Gale Group. All rts. reserv.

03240360 Supplier Number: 44453387 (USE FORMAT 7 FOR FULLTEXT)
MOTOROLA READIES DSP DESIGN KIT

Electronic News (1991), p10
Feb 21, 1994
Language: English Record Type: Fulltext
Document Type: Magazine/Journal; Trade
Word Count: 836

... platform - which later this year will be boosted by a new scaler 60Mips DSP.

The PC Media developer's kit initially targets OEMs with an **ISA** -based subsystem that replaces several add-in cards. Single boards based on the PC Media technology, priced at less than $300, will support MPEG audio ...

...standards.

The developer's kit is priced at $7,500; Motorola's current offering is a host-based, single-tasking DSP operating system supporting the **ISA** bus and Windows 3.1. In addition to the DS056002, Motorola is making available a low -power DSP for portable systems, the DSP56L002.

Motorola's DSP Division, Microcontroller Technologies Group developed its own message protocols and drivers internally, using **assembly** language **code** for the various DSP tasks. Media PC will support Microsoft's Windows Telephony Application Program Interface (TAPI) and Microsoft At Work standards.

While Motorola expects...

...and EPM5128LC interface.

Available in Q2, the PC Media developer's kit includes reference board design schematics, demonstration task software, demonstration application software, PLD equations, **assembler** /linker, a C **compiler** , a **command** converter and documentation.

The PC Media platform will include third-party software from Centigram Communications, DSP Group, Motorola Codex, QSound Labs and Peavey Electronics. When...

...require simultaneous data/voice conferencing.

Motorola's new scalar 60Mips DSP, code-named Onyx, will sample later this year. Unlike the semi-scalar DSP56002, the **new** Onyx DSP will execute one **instruction** per clock cycle. The Onyx DSP will operate at 3.3V and use a 0.5 micron process; first silicon is expected in Q4 with...

14/3,K/13 (Item 1 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

12056928 SUPPLIER NUMBER: 61872548 (USE FORMAT 7 OR 9 FOR FULL TEXT)
dsp directory 24 bits.(Motorola DSP563xx)(Product Information)
EDN, 45, 7, 93
March 30, 2000
ISSN: 0012-7515 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 761 LINE COUNT: 00066

... and data buses. The DMA transfers data among memories (P, X, and Y) or among memory and peripherals or the external host buses (PCI or **ISA**). You can program the size of the program RAM, instruction cache, X-data RAM, and Y-data RAM.

The static core operates from dc to...

...built-in prescaler that allows dynamic clock throttling. For additional power savings, the core automatically powers down unused memories, peripherals, and core logic on every **instruction** .

The **newest** members of the 56300 family are the 56307 and 56311. These devices include an on-chip enhanced filter coprocessor (EFCOP) that processes filter algorithms in...

...an application-development system to evaluate the chip and debug target systems. The system comes with an application-development module, a host-interface card, a **command** converter, an **assembler** , simulator

software, and a C compiler. The 563xx's JTAG-based OnCE port allows you to examine all internal buses in real time and record...

14/3,K/14 (Item 2 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

12056927 SUPPLIER NUMBER: 61872547 (USE FORMAT 7 OR 9 FOR FULL TEXT)
dsp directory 16 bits.(Buyers Guide)
EDN, 45, 7, 62
March 30, 2000
DOCUMENT TYPE: Buyers Guide ISSN: 0012-7515 LANGUAGE: English
RECORD TYPE: Fulltext
WORD COUNT: 16676 LINE COUNT: 01348

... instruction to indicate the end of the loop and to perform the ... counter decrement.
Support--LSI offers a Gnu-based compiler, a linker, and an **assembler**. The compiler supports new C fixed-point **data** types and employs a variety of C-intrinsic functions. Development platforms include an integrated debugging environment, flash, RS-232C and JTAG-based host communication and...

...architecture is its variable-length-execution-set (VLES), or explicitly parallel instruction computing (EPIC), model that promotes scalable resources (such as multiple ALUs), a scalable **instruction - set architecture** , and increased orthogonality. Traditional very-long-instruction-word (VLIW) architectures use fixed-length instructions in a fixed-length execution set. This architecture sometimes requires the **compiler** to insert nonoperating **instructions** to fill unused instruction slots, and the instructions and execution sets have memory-alignment restrictions. On the other hand, VLES allows variable-length instructions with...

...To further increase flexibility and scalability, StarCore uses the prefix "construct" to add features to instructions. You may add in one or two prefixes per **instruction** to accommodate **additional** registers and conditional execution. The StarCore architecture can also group multiple instructions into execution sets, which the architecture executes simultaneously. Two 64-bit data buses...

...40 contains 16 40-bit registers and 27 32-bit address registers.
Another key StarCore feature, the associated compiler, can detect parallelism and group-independent **instructions** . The SC140's **compiler** statically builds execution sets comprising one to six **instructions** . Relying on the **compiler** to encode the parallelism minimizes the silicon resources for **instruction** decoding and dispatching. The **compiler** performs a variety of optimizations, including software pipelining, function inlining, if conversion to exploit predicated execution, global scheduling, and sophisticated loop analysis and transformations to...
...the DSPs provide no architectural support, or addressing modes using the stack are less efficient than those using absolute addressing. The SC 140 compiler implements " **compiled** stacks" to provide the **functions** of stack accesses at the cost of absolute addressing, thus avoiding extra runtime overhead. The SC140 compiler also implements DSP-memory-specific optimizations, such as...

14/3,K/15 (Item 3 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

11484915 SUPPLIER NUMBER: 56950234 (USE FORMAT 7 OR 9 FOR FULL TEXT)
TENSILICA XTENSA.
EDN, 44, 19, 176
Sept 16, 1999
ISSN: 0012-7515 LANGUAGE: English RECORD TYPE: Fulltext

WORD COUNT: 1040 LINE COUNT: 00089

... This bit holds a tag that indicates whether the corresponding bit of AR contains valid data. The use of speculative loads and the AV register **file** allows the compiler to move loads and dependent instructions earlier in the code, but these instructions have the same exception characteristics as nonspeculative instructions.

From...

...space.

After you profile your application and determine the location of the software bottlenecks, you can use Tensilica's Instruction Extension Language (TIE) to define **new** mnemonics, opcodes, and associated computational functions. TIE is a subset of Verilog and integrates with the rest of the processor RTL to allow Tensilica's synthesis tools to automatically implement the **new instructions** in the hardware and in the compiler/simulator suite. TIE requires that the **new instructions** access only register-based **operands** , but you can define **additional** registers of arbitrary width to enable computations in protocol-, image-, and signal-processing tasks. Another restriction is that all **new instructions** must execute in one cycle to match the timing of the base processor.

The base processor does not support exceptions or interrupts, and, depending on...

...select from several Tensilica-provided DSP-instruction-extension packages or single-in-struction, multiple-data instructions. The MAC16 DSP option, for example, adds about 70 **new instructions** to the core **instruction** -set **architecture** . In addition to the standard integer, branch, and logical instructions, the base architecture supports conditional moves, a variety of shifts, and zero-overhead loop instructions. The loop **instructions** effectively perform loop index update, loop comparison, and branch in parallel with any other instruction.

Development tools: Tensilica's tool suite comprises a Gnu-based ANSI C/C++ compiler, an **assembler** /dissassembler, a debugger, a linker, a code profiler, and an instruction simulator. Tensilica also provides function libraries and is working with Wind River Systems (www...

14/3,K/16 (Item 4 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB
(c)2004 The Gale Group. All rts. reserv.

11484896 SUPPLIER NUMBER: 56950214 (USE FORMAT 7 OR 9 FOR FULL TEXT)
ZILOG Z80/80180/80380.
EDN, 44, 19, 140
Sept 16, 1999
ISSN: 0012-7515 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 678 LINE COUNT: 00058

... search instructions support automatic address incrementing and decrementing. The 380 instruction set includes mode settings; instruction prefixes for identifying 16- or 32-bit data; and **new instructions** that control whether load, store, and arithmetic instructions operate on 16- or 32-bit data.

Special on-chip peripherals: Z18x devices beyond the Z80180 include ...

...timer from the 180 family, plus eight advanced DMA channels, three high-speed data-link-control channels, a PCMCIA interface, and a plug-and-play **ISA** interface.

Development tools: Except for one low-cost Z80180 emulator, Zilog relies on third parties for emulation support for the Z80, Z8018x, and Z8038x devices...
...com). Zilog supplies several evaluation boards and a low-cost emulator. Unlike with the Z180, development support for the Z380 is minimal; Zilog provides an **assembler** , an evaluation board, and a Production Languages Corp (www. plcorp.com) C **compiler** with an optimizer **program** to improve

performance and code size. Microtec (www.microtec.com) offers a Z80380 C compiler.

Second sources: Seven licensed vendors act as second sources for...

14/3,K/17 (Item 5 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB

08124425 SUPPLIER NUMBER: 17389671 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Plastics technology: manufacturing handbook & buyers' guide 1995/96.(Buyers Guide)
Plastics Technology, v41, n8, pCOV(941)
August, 1995
DOCUMENT TYPE: Buyers Guide ISSN: 0032-1257 LANGUAGE: English
RECORD TYPE: Fulltext
WORD COUNT: 174436 LINE COUNT: 15187

... hr.

Three types of motionless mixers improve homogeneity of the melt stream in injection molding machines and extruders: LPD for lower viscosities where pigments are **added** , ISG for high viscosities, and Blendex-modified ISG for batch or continuous solids/solids blending.

SCOTT TURBON MIXER, INC.

High-shear and low-shear, batch...the hollow unit uniformly cools the end of the blow pin and evenly cools necks of blow molded bottles. More than 350 sizes stocked. (See **data** sheet p. 271.)

OGDEN MANUFACTURING CO.

Mighty Watt and Mighty Watt Plus cartridge heaters. Mighty Watt Plus allows thousands of combinations of size, rating, and...heat distribution on uniform and irregular surfaces where moderate heating is required.

Complete engineering and R&D services using CAD available for custom-designed heating **assembly** needs. (See also Hot-Runner Components and Systems.)

THE SHUMAN CO.

Thermoforming machinery supplier offers its own quartz-tube infrared heater panels. Elements are completely...loop control, and power on/off switch. With microprocessor PID control and complete diagnostics, module operates automatically after setting temperature and turning on power. CompuStep **function** provides automatic warm-up, and when complete, controller switches to CompuCycle, a zero-crossover power drive that delivers smooth, continuous a-c power, which reportedly...

14/3,K/18 (Item 1 from file: 810)
DIALOG(R)File 810:Business Wire

BW0691

CYGNUS: Cygnus Announces GNUPro Support for the MIPS16 ISA and the LSI Logic 16/32-Bit TinyRISC Microprocessor

October 21, 1996

Byline: Business Editors/Computer Writers

MOUNTAIN VIEW, Calif.--(BUSINESS WIRE)--Oct. 21, 1996--Cygnus Support today announced availability schedules for its industry-leading GNUPro software development tools for the MIPS16 **Instruction Set Architecture** from MIPS Technologies, Inc. and the LSI Logic's 16/32-Bit TinyRISC microprocessor.

These tools will allow LSI Logic customers using Cygnus tools to...

...Barton, director of marketing for Cygnus' GNUPro products. "We enable this code-size reduction by optimizing the GNU compiler technology to take advantage of the **new** MIPS16 **instruction set architecture** jointly developed by LSI Logic and MIPS Technologies Inc."

The Cygnus GNUPro port to the TinyRISC MPU will originate from the same, single source tree...

...well-known in the industry," he added.

The Cygnus GNUPro toolchain consists of the GNU C/C++ compilers, remote source-level debugger with GUI, macro- **assembler** , linker/loader, libraries, binary **file** utilities, and instruction set simulator. Cygnus offers flexible and comprehensive support services for its GNUPro products.

12/9/27 (Item 4 from file: 148)
DIALOG(R)File 148:Gale Group Trade & Industry DB

03924948 SUPPLIER NUMBER: 07336142 (THIS IS THE FULL TEXT)
Software tools handle all (microprocessor) traffic: universal cross-assemblers. (technical)
Leibson, Steven H.
EDN, v34, n12, p89(7)
June 8, 1989
DOCUMENT TYPE: technical ISSN: 0012-7515 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2693 LINE COUNT: 00218

ABSTRACT: For engineers who routinely work with several types of processors, universal **cross - assemblers** can solve such problems as having to buy a new assembler for each design. Although universal, or retargetable, **cross - assemblers** do not comply precisely with a microprocessor's assembly language, for a wide variety of processors they can translate assembly-language source code into machine code. They also permit engineers to quickly develop subsets of standard processor **instruction sets** and to define one processor's **instruction set** using another processor's mnemonics. Commercially available universal **cross - assemblers** , such as AnyWare Engineering's CASM, are discussed in detail.
TEXT:

Software tools handle all [mu]P traffic

If you routinely work with several types of processors, you're no doubt disenchanted with buying a new assembler for each of your designs. Furthermore, the added cost of yet another assembler may tilt the economic scales against using the "perfect" [mu]P for a project, obliging you to choose an older or less efficient device because you already own the software-development tools for that processor. Universal cross-assemblers can help you solve these dilemmas permanently.

Although universal (also called retargetable) **cross** -assemblers don't provide exact compliance with a [mu]P's assembly language, they can translate assembly-language source code into machine code for a wide variety of processors. They also let you quickly develop subsets of standard processor **instruction sets** . You might use this capability, for example, to write code for an ASIC processor core you optimized (shrunk) by removing the hardware that executes unnecessary instructions.

You can also use a universal **cross - assembler** to define one processor's **instruction set** using another processor's mnemonics. With this feature, you can convert a program written for one processor into another's machine code rather painlessly.

Despite the multiple benefits of universal **cross assemblers** , vendors that provide assemblers for standard [mu]Ps and [mu]Cs generally don't sell universal **cross - assemblers** . Instead, they offer a range of separate assemblers closely tailored to an individual processor or processor family. For example, Boston Systems Office (Waltham, MA) offers a wide variety of tailored assemblers. It even sells more than one assembler for the Motorola 6800 [mu]P family because the **instruction sets** differ slightly between the 6800 and Hitachi's compatible 6300 [mu]P series. Several other third-party vendors of assemblers for standard processors, including Enertec Inc (Lansdale, PA), Introl Corp (Milwaukee, WI), and Lear Com Co (Lakewood, CO), offer assemblers suited to specific processors.

Most assembler vendors don't feature universal assemblers because they believe that they can't make an assembler that recognizes any assembly-language statement. Assembler directives (pseudo ops) and numbering schemes (such as writing the hexadecimal equivalent of 56 as 56H or $56) present the major stumbling blocks in developing a truly universal cross-assembler. The assembly language for Intel's 80X86 [mu]P family, for example, includes a huge number of assembler directives and pseudo ops. Most of these assembler directives are specific to Intel's [mu]Ps; you won't find many analogous pseudo ops in assemblers from other vendors, unless the assemblers are specifically designed to accept Intel's assembly code.

Designing an assembler that conforms with just the instruction-set syntax, however, doesn't seem to present much of a problem. In fact,

assembler vendors often use universal assembler generators to perform the majority of work in creating tailored assemblers. They can then make minor adjustments and fine tune the assemblers by hand.

You may interpret this practice as a blatant attempt to improve sales by keeping the cost-effective technology in house. However, vendors of tailored assemblers cite the same reason for withholding universal products: Many customers want assemblers that exactly conform to the [mu]P or [mu]C vendor's assembly-language syntax. In some cases, this strict compliance is warranted.

Assemblers that comply with the semiconductor vendor's original assembly language let you reuse code that you or your company previously wrote using original-equipment assemblers. In addition, tailored assemblers let you easily assemble source-code routines that you obtain from other sources. You may also need this strict compliance to assemble source code generated by a high-level-language compiler.

Some engineers, however, don't need to preserve existing assembly-language source code, and many of these same engineers write assembly-language programs for a large number of [mu]Ps and [mu]Cs. For these people, the single set of assembler directives and the standard command syntax of a universal cross-assembler represent advantages instead of liabilities. If you don't need the exact compliance of a tailored assembler, if you routinely work with several different processors, and if you don't have a budget to buy software tools at whim, then you too can benefit from using a universal cross-assembler.

Similar needs and considerations prompted three engineers to develop several of the universal cross assemblers that are now available (Table 1). Although they initially wrote these programs for their own use, these people later offered the programs to other engineers and eventually introduced their programs and software-development packages to the commercial market.

In 1985, for example, Thomas Anderson developed TASM (table-driven assembler) because he tired of hand assembling [mu]P programs for products he designed to aid blind people. Anderson's designs incorporate Texas Instruments' speech chips, giving voice to instruments such as a glucometer (which measures blood-sugar levels), a dietary scale, a tachometer, and a skin caliper. After testing his universal assembler in commercial waters, Anderson decided to market TASM as shareware through his company, Speech Technology.

Because it's a shareware product, you can obtain TASM at a low cost. PC-SIG (Sunnyvale, CA) provides a copy for $6 on its disk #643, or you can download the program from the Library Bulletin Board System (Seattle, WA). You can also find TASM on many information services, such as Compuserve, but those sources may not have the latest version of the assembler.

You can also obtain a registered copy of TASM directly from Speech Technology. Anderson requests a $30 registration fee for the package. Registered users receive the latest version of the assembler and are entitled to support.

Speech Technology supplies TASM with several instruction-set definitions (Table 2). You can also write instruction-set definitions for TASM by creating an instruction-set table with a text editor. In fact, one of the tables included in TASM version 2.7 is an instruction-set table. It accommodates the Zilog Z80 [mu]P and was created by Carl A Wall, one of TASM's users.

Each line of a TASM instruction-set table consists of six fields. The first five fields in the table define the instruction mnemonic, the instruction arguments (if any), the 1-byte op code that is represented by the mnemonic, the size of the instruction (in bytes), and the "MODOP" field, which performs a variety of operations on the instruction and argument bytes such as swapping bytes or merging. The op code's 1-byte size confines the assembler to 8-bit processors.

The sixth field in the table defines an instruction's class. TASM lets you define classes (sets) of instructions and designate which class the program should use when you assemble a file. You can use this feature to define an **improved** processor's extended **instruction set** . The TASM table for the 6502 [mu]P, for example, defines **extra instructions** for Rockwell's R65C00 and R65C02 microprocessors.

For reasons resembling Anderson's, Jonathan Griffitts wrote a universal **cross - assembler** called CASM to help him with his consulting

work. Griffitts has been using his assembler since early 1987 to assemble code for the many different [mu]Ps and [mu]Cs. He now markets CASM through his company, AnyWare Engineering.

CASM can assemble code for 8-, 16-, and 32-bit processors and includes instruction-set definitions for many [mu]Ps and [mu]Cs. If you want to create an instruction-set definition for a processor not supported by AnyWare, you must create a text file containing that definition, written in AnyWare's proprietary instruction-set definition language. You must then compile your definition using the package's DEFCOMP compiler, which produces a control file for the assembler.

CASM can accommodate some fairly complex processor instruction sets because it uses a definition language that includes C-like constructions such as subroutines, loops, and user-defined data types, instead of simple instruction-set definition tables. The instruction-set definition for NEC's 7720 DSP [mu]C, which AnyWare supplies in the CASM package, illustrates the flexibility of this definition language. This unusual DSP [mu]C features a 23-bit instruction word and a 13-bit-wide ROM for storing constants that can be difficult or impossible to describe within the structured environment of a table.

A third engineer, Peter Aske, also developed a universal cross-assembler to simplify his own work. While working as an engineer in Nova Scotia, he found that he needed development software to write code for several different [mu]Ps. In addition to the cost of buying a new assembler for each of his designs, Canadian import duties and an unfavorable monetary-exchange rate made purchasing commercial assemblers from US vendors unattractive. As a result, Aske produced a universal cross-assembler using Borland International's Turbo Pascal compiler. On a friend's recommendation, he transformed Cross-8 into a commercial product.

Cross-8 became the first **cross - assembler** marketed by Aske's company, Universal **Cross - Assemblers** . Because the program was written in Turbo Pascal, versions of Cross-8 were available for Digital Research's CP/M and Microsoft Corp's MS-DOS operating systems. Cross-8 is now obsolete and has been succeeded by two more powerful products: Cross-16 and Cross-32. As their names imply, these products assemble code for 16- and 32-bit processors as well as for processors with smaller instruction-word sizes. Both assemblers employ a multipart **instruction - set** table to define the processor instructions. The **instruction - set** tables are stored in text files, so you can use a text editor to modify an existing table or create a new one.

Cross-16 and Cross-32 are written in C, which makes them portable across a variety of computers. Universal **Cross Assemblers** offers its assemblers for use with the MS-DOS operating system. Macrochip Research, however, licensed the source code for Universal Cross Assemblers' Cross-32 and offers the assembler as part of a universal code-development package called the Macrochip Development Environment. The vendor markets this package for use on several computers that run different operating systems.

The Macrochip Development Environment includes a universal cross-assembler, a text editor for creating source code, and a communications program for transferring assembled object code to your target hardware through an in-circuit emulator. The various components of the package are tied together by a menu-driven user interface. Macrochip Research also sells in-circuit emulators for various [mu]Ps and [mu]Cs and offers the Development Environment as a support tool for its emulators.

Like Macrochip Research, Stag Microsystems offers a universal cross assembler as part of a comprehensive software package--the Stag VSDS (versatile software-development system) package. Stag's universal cross-assembler is an ideal companion for its EPROM emulator. You can use the general-purpose emulator hardware to develop code for systems built around any processor, as long as your target system has an EPROM socket.

In addition to its universal cross-assembler, the VSDS package incorporates several other programs, including a text editor, a linker, a universal disassembler, a "make" facility, an instruction-table generator, and a communications utility that transfers code to the EPROM emulator. Stag also offers the SDS package, which includes the software from the VSDS package without the EPROM emulator. Even if you don't buy Stag's emulator, you still need a slot in your computer that accommodates the EPROM emulator's interface card because it also serves as a software antipiracy device.

Like linkers used with tailored assemblers, the VSDS linker binds object modules together, resolves address references between these modules, and emits an absolute object file. Of the universal cross-assemblers listed in Table 1, only the products from Stag and AnyWare Engineering include linkers. The remaining products generate absolute object code directly from the assembler. If you're writing large programs or have more than one programmer working on your project, you may find that the combination of an assembler and a linker lets you break your assembly-language program into manageable chunks.

You can achieve a similar effect with "include" directives, which tells the assembler to merge text stored in separate source files during the assembly. All the assemblers listed in Table 1 have the "include" capability. Even if you use these statements, you must reasemble your entire program after modifying any piece of it. Reassembling an entire program takes more time than assembling just one source-code module and relinking the program. As a result, assemblers that generate relocatable code are often more efficient for building large programs than are absolute assemblers.

The VSDS package's "make" facility automatically helps you avoid reassembling all your source-code modules each time you alter a program. After you specify your program's modules in a text file called the "makefile," the "make" facility reassembles only the program source files that have a later date and time stamp than their corresponding object files. The "make" facility won't reassemble any files that you haven't edited since the program was last assembled.

You should also note the package's universal debugger, which lets you set breakpoints, read and alter register contents, and examine and modify information stored in your target system's RAM. The debugger resides in a personal computer running the VSDS package and works with any processor and instruction set by reading the same instruction-set tables created for the VSDS assembler and disassembler. To make the universal debugger work with your target system, you must write a small amount of monitor code that runs in your target system and interacts with the debugger in the PC.

The monitor software communicates with the PC running the debugger over an RS-232C link. You need to dedicate a serial port in your target system for this purpose, at least during the debugging process. Stag provides a prototype version of the monitor code, written in the assembly language for Motorola's 6800 [mu]P. You can use this code as a model when writing your own code. The monitor software you write must be able to transmit and receive information via the serial port and must respond to commands generated by the debugger in the PC. Monitor routines typically require about 200 to 300 types of code.

Whichever universal product you choose to incorporate in your designs, you needn't limit it to the general purpose of assembling code for your [mu]P. You can also use these software-development tools to simplify related design tasks. For example, you can use a universal **cross - assembler** to standardize programs with one assembly-language style or one **instruction set** for every processor you use. This feat is simple to perform with closely related processors, such as the Zilog Z80 and Intel [mu]Ps. If you like the Zilog Z80 [mu]P's code syntax and argument structure (move source to destination) better than Intel's syntax for the 8080 [mu]P (load destination from source), for example, you can extend Zilog's format to the Intel processor with the aid of a universal **cross - assembler** . Similarly, if you prefer Intel's syntax, you can extend it to Zilog's processors. You can also use a universal **cross - assembler** to create one assembly-language syntax for several unrelated processors.

Further, you may want to use a particular [mu]P or [mu]C, but discover that an assembler isn't yet available for that processor either because the device is too new or because the expected market for the processor is too small to catch the tailored-assembler vendors' attention. Or, you may dislike the processor's original-equipment assembler. In either case, you can create the program you need if you add a universal cross-assembler to your software-development tool kit.

```
Set     Items   Description
S1    5452704   PROCESSOR? OR MICROPROCESSOR? OR COMPUTER? OR MICROCOMPUTE-
            R? OR (CENTRAL()PROCESSING OR CONTROL OR MICRO?)()(PROCESSOR?
            OR COMPUTER? OR UNIT? OR DEVICE?) OR DATA()COLLECTION()DEVICE?
S2    2500597   SOURCE()(CODE OR CODES) OR STATEMENT? OR ASSERTION? OR DEC-
            LARATION? OR INSTRUCTION?
S3    5301710   COPYING OR COPY OR COPIES OR STORE? ? OR STORING OR SAVE OR
             SAVING OR KEEP OR KEEPING OR PRESERV? OR MEMORY OR CACHE OR -
            CACHING OR CACHED OR CACHES OR BACKUP OR BACK()UP
S4       1838   (TEMPORARY OR IMPERMANENT OR INTERIM OR PROVISIONAL OR SHO-
            RT()(TERM OR RANGE))()(FILE OR FILES)
S5   12517941   APPLY OR APPLIED OR APPLIES OR USE OR USES OR USING OR UTI-
            LIZ? OR EMPLOY? OR IMPLEMENT? OR EXECUT?
S6      84919   ASSEMBLER? OR LINKER? OR COMPILER? OR LANGUAGE()PROCESSOR
S7   11390367   PRODUCE? OR GENERATE? OR REPRODUCE? OR CREATE? OR DEVELOP?
S8      52877   (OBJECT OR MACHINE)()(CODE OR CODES) OR ASSEMBLY()LANGUAGE-
            ()SOURCE()CODE OR ASSEMBLY()LANGUAGE()INSTRUCTION OR ISA
S9         46   S2 (S) S3 (S) S4
S10         7   S2 (S) S4 (S) S5 (S) S6
S11      1286   S6 (S) S7 (S) S8
S12         0   S9 (S) S11
S13        77   S2 (S) S4
S14         0   S13 (S) S8
S15        24   S9 (S) S1
S16         2   S10 (S) S1
S17        28   S13 (S) S1
S18       450   S11 (S) S1
S19       200   S18 (S)  S2
S20        77   S19 (S) S3
S21         0   S20 (S) S4
S22         0   S9 (S) S11
S23       110   S10 OR S15 OR S17 OR S20
S24        93   S23 NOT PY>2000
S25        93   S24 NOT PD>20001222
S26        64   RD (unique items)
File  15:ABI/Inform(R) 1971-2004/Jan 29
         (c) 2004 ProQuest Info&Learning
File 810:Business Wire 1986-1999/Feb 28
         (c) 1999 Business Wire
File 647:CMP  Computer Fulltext 1988-2004/Jan W3
         (c) 2004 CMP Media, LLC
File 275:Gale Group Computer DB(TM) 1983-2004/Jan 29
         (c) 2004 The Gale Group
File 674:Computer News Fulltext 1989-2004/Jan W4
         (c) 2004 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2004/Jan 15
         (c) 2004 The Dialog Corp.
File 624:McGraw-Hill Publications 1985-2004/Jan 28
         (c) 2004 McGraw-Hill Co. Inc
File 636:Gale Group Newsletter DB(TM) 1987-2004/Jan 29
         (c) 2004 The Gale Group
File 813:PR Newswire 1987-1999/Apr 30
         (c) 1999 PR Newswire Association Inc
File 613:PR Newswire 1999-2004/Jan 29
         (c) 2004 PR Newswire Association Inc
File  16:Gale Group PROMT(R) 1990-2004/Jan 29
         (c) 2004 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 553:Wilson Bus. Abs. FullText 1982-2004/Dec
         (c) 2004 The HW Wilson Co
```

26/3,K/2 (Item 2 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

02049139 56590885
Programming languages
Kay, Russell
Computerworld v34n29 PP: 64 Jul 17, 2000
ISSN: 0010-4841 JRNL CODE: COW
WORD COUNT: 839

...TEXT: code than other languages.

Different Strokes

But how could you make a program run on two different **computers** ? You would have to reprogram it for the second machine's **instructions** . The answer was a higher-level language that could be adapted to different **computers** by processing the application code through another program, known as a compiler, which translated the application into machine code and **stored** it as a file. Or it could be done through an interpreter, which did the same job as a **compiler** on the fly and then ran the program without **storing** it. The **compiler** or interpreter was specific to each **computer** type, but it needed to be **created** only once and could then process many programs written in the high-level language. The first important...

26/3,K/4 (Item 4 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

00061144 77-13539
THE MICROCOMPUTER DEVELOPMENT SYSTEM
CASSELL, DOUGLAS A.; CAVANAUGH, JOHN E.
MINI-MICRO SYSTEMS V10 N8 PP: 34-36, 38, 40 AUGUST 1977
JRNL CODE: MOD

ABSTRACT: ALL **MICROCOMPUTER DEVELOPMENT** SYSTEMS INCLUDE CONSOLE CONTROLS TO START THE OPERATING SYSTEM SOFTWARE, A KEYBOARD FOR PROGRAM PREPARATION AND INPUT...

... AND AN INPUT DEVICE SUCH AS A CASSETTE TAPE TO READ SOURCE AND OBJECT PROGRAMS INTO THE **COMPUTER** WITHOUT BEING RETYPED. A PROM PROGRAMMER IS ANOTHER PERIPHERAL THAT MAY BE USED. HANDWRITTEN **SOURCE CODE** IS KEYED INTO AN ON-LINE TERMINAL IN CONJUNCTION WITH A TEXT EDITOR PROGRAM. A **COMPILER** OR **ASSEMBLER** THEN TRANSLATES THE **SOURCE CODE** TO **OBJECT CODE** . A LOADER IS THEN USED TO READ THE OBJECT PROGRAM INTO **MEMORY** . A LINKING LOADER MAY BE USED TO COMBINE MORE THAN ONE PROGRAM INTO A SINGLE PROGRAM FOR DIRECT EXECUTION BY THE **COMPUTER** . PROGRAMS UNDER TEST ARE THEN RUN UNDER CONTROL OF A DEBUGGER PROGRAM WHICH ALLOWS THE PROGRAMMER TO ...

26/3,K/5 (Item 1 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext

01226776 CMP ACCESSION NUMBER: EET20001113S0085
Compiler extensions alter AltiVec
Kalpesh Gala, Product Manager, Motorola Inc., Austin, Texas, Mike Haden, Compiler Development Manager, Green Hills Software Inc., Santa Barbara, Calif.
ELECTRONIC ENGINEERING TIMES, 2000, n 1140, PG122
PUBLICATION DATE: 001113
JOURNAL CODE: EET LANGUAGE: English
RECORD TYPE: Fulltext

SECTION HEADING: COMMUNICATIONS - FOCUS: MULTIMEDIA NETWORKS
WORD COUNT: 1579

... unroll loops in this manner , or to even greater depths, subject to explicit direction from the programmer.

Saving execution cycles is one of the goals of optimization, and using the right data structure can advance...

...compiler can separate complex vectors into two sets, allowing for faster handling. Many compilers allow you to **save** cycles by rearranging **object code** to hide data load latency. For example, by interleaving load and add **instructions** , the **processor** can perform one operation while data is loading for another. Having the **compiler** automatically **generate** such code patterns saves programmers from taking the time to make such optimizations by hand.

Another optimization...

26/3,K/6 (Item 2 from file: 647)
DIALOG(R)File 647:CMP Computer Fulltext

01022124 CMP ACCESSION NUMBER: WIN19940601S1855
TESTING, TESTING
Ed Curry
WINDOWS MAGAZINE, 1994, n 506 , 218
PUBLICATION DATE: 940601
JOURNAL CODE: WIN LANGUAGE: English
RECORD TYPE: Fulltext
SECTION HEADING: Features

TEXT:
... Cx486DX/50GP. (IBM did not make a 486 chip with built-in math coprocessor.) We chose these **processors** because they could be installed in the same systems, minimizing differences across test runs. Intel, AMD and Cyrix voluntarily provided the latest versions of their **processors** for the study. (Cyrix did not have a 66MHz CPU in its line, but company officials agreed...

...tests target the key functions of the 486 CPU, including error and exception handling, floating-point calculation, **memory** management, multitasking and **memory** protection and the CPU's ability to emulate earlier Intel **processors** (such as the 8088 and 80286). The tests run so far have concentrated on numeric processing, so...

...C, C++, Pascal and Fortran and is designed to be representative of real-world software. We used **compilers** from Borland, IBM, MetaWare, Microsoft, SCO and Watcom to compile the code, using a variety of **processor memory** models. Test hardware consisted of nine custom-built systems that used identical components (except for the CPUs), including the NICE Green VL/ **ISA** motherboard from Lion **Computer** . We chose this board because it included a ZIF (zero insertion force) socket and could be configured to accommodate some special features of the Cyrix CPU. Each system was configured with 256KB of **cache** and 16MB of RAM. Fujitsu provided SCSI hard disks, allowing each test system to contain 3GB of...

...executed the tests to rule out an error in data collection. 2. We replaced the CPU that **produced** the incorrect results with another CPU of the same make and type, and reran the tests. 3. We removed the suspect CPU from the system and replaced it with an Intel **processor** to eliminate the possibility of faulty system hardware. 4. We installed the suspect CPU in a system from which the Intel **processor** was removed and ran the tests a final time. Any results that still failed to match Intel...

...date. (Results were valid at the time this article was written.)

Surprisingly, although the AMD and Intel **processors** ran at the same speeds, the Am486 CPU consistently outperformed Intel's 486. AMD insisted we add...

...operating temperature. To ensure a level playing field, we added cooling fans to all three brands of **processors** . During testing, one of the AMD CPUs demonstrated an unusual failure. Each series of floating-point tests....

...of a post- shipment failure. When we replaced the faulty CPU with a new Am486, the new **processor** **produced** consistently correct results. That leads to the good news: After installing the new AMD CPU and re-executing the tests, AMD's 486 **processor** has, to date, demonstrated 100 percent Intel compatibility. In every test executed, the AMD results were identical to those **produced** by the Intel 486DX2. (While we **preserved** the faulty part and its results for later analysis, we did not include its responses in the...

...at 50MHz, since Cyrix could not provide a clock-doubled 33/66MHz for the study. Cyrix's **processor** had two serious problems. First, it refused to interact with the RAM **cache** **memory** installed on the motherboard. The **cache** **memory** is one of the few components in the system with unbuffered access to the CPU. When the system external **cache** was enabled, the Cyrix CPU would either hang the system or **generate** a CPU exception that halted the system. Once the external **cache** was disabled, the Cyrix 486 would allow the system to boot the operating system. Neither the AMD nor the Intel **processors** demonstrated this problem. The second and more troubling problem concerned the floating point. Under OS/2 2.1, the Cyrix **processor** failed 9 of 238 floating-point compatibility tests. These failures ranged from reporting incorrect results to reporting **instructions** . Another divergence between the Cyrix and Intel (or AMD) CPUs was in its rounding of floating-point...

26/3,K/10 **(Item 4 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02449021 SUPPLIER NUMBER: 66889798 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Compiler extensions alter AltiVec. (Product Announcement)
Gala, Kalpesh
Electronic Engineering Times, 122
Nov 13, 2000
DOCUMENT TYPE: Product Announcement ISSN: 0192-1541 LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 1718 LINE COUNT: 00146

... unroll loops in this manner, or to even greater depths, subject to explicit direction from the programmer.
Saving execution cycles is one of the goals of optimization, and using the right data structure can advance...

...compiler can separate complex vectors into two sets, allowing for faster handling. Many compilers allow you to **save** cycles by rearranging **object** **code** to hide data load latency. For example, by interleaving load and add **instructions** , the **processor** can perform one operation while data is loading for another. Having the **compiler** automatically **generate** such code patterns saves programmers from taking the time to make such optimizations by hand.
Another optimization...

26/3,K/14 **(Item 8 from file: 275)**
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

02124781 SUPPLIER NUMBER: 20048572 (USE FORMAT 7 OR 9 FOR FULL TEXT)

OPEN GROUP TOUTS HIGH-SPEED JAVA INSTALLATION COMPILER.
Computergram International, n3306, pCGN12050004
Dec 5, 1997
ISSN: 0268-716X LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 549 LINE COUNT: 00045

TEXT:
The Open Group claims a Turbo-J **compiler** it will begin selling within 60 days for use Hewlett-Packard Co's PA-RISC **computers** overcomes some of the limitations of Java JIT just-in-time **compiler** technology by enabling users to **store** Java applications as highly-optimized native executables or dynamically loadable shared libraries that do not need to...

...the network. Turbo-J compiles machine-independent Java byte-code (zeros and ones rather than human-readable **source code**) into **machine code** which can run faster than interpreted byte-code because it is optimized for the system's specific or native **instruction** set. Unlike JITs which compile whole applications on the fly, Turbo-J optimizes only performance-critical parts...
...Also, because Turbo-J requires the system to perform complex optimization tasks on the entire program the **compiler** works off-line, generating code in a few minutes rather than in a few seconds like JITs...

...up to 100 times faster than standard byte code runs. HP provided most of the funding for **development** of Turbo-J which is why it's up on HP-UX, although versions of Linux-on...

...for Wind River Systems Inc VxWorks embedded operating system are due next quarter. (Open Group is currently **developing** two versions of an embedded J-Lite Java execution environment for Pentium and PowerPC; one incorporates the...

...a next-generation implementation has already begun. Open Group is positioning Turbo-J as a native installation **compiler** for the type of Java bytecode applications it thinks will soon become popular with ISVs. Instead of...

...and waiting for JIT compilation each time the spreadsheet is used Turbo-J compiles it once and **stores** in natively for re-use. Open Group says Turbo-J has evolved from its long-gestating Architecture...
...s SuperCede for Windows is similar to Turbo-J and well as Inria's Harissa native Java **compiler** although the latter uses its own runtime, where Turbo-J uses the loader, **memory** allocation, garbage collection and multi-threading of the platform's native JDK.

26/3,K/15 (Item 9 from file: 275)

02117712 SUPPLIER NUMBER: 19912657 (USE FORMAT 7 OR 9 FOR FULL TEXT)
AMD picks DEC at MPU Forum. (AMD announces at the Microprocessor Forum that it will use DEC's 21264, or EV6, bus for its next-generation microprocessors) (Company Business and Marketing)
Electronic News (1991), v43, n2190, p4(2)
Oct 20, 1997
ISSN: 1061-6624 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 881 LINE COUNT: 00089

ABSTRACT: AMD announced at the **Microprocessor** Forum in San Jose, CA, in Oct 1997 that it will use DEC's 21264 (EV6) bus in its next-generation **microprocessors** . Intel showed its Merced chip, which it is developing with HP, and described the architecture Merced and other chips will use as the Explicitly Parallel **Instruction** Computing (EPIC) architecture. Intel Dir of **Microprocessor** Architecture John Crawford said current **microprocessor** performance is limited by **memory** latency relative to **processor** speed and load delay compounded by machine width. Crawford said explicit

parallelism has the compiler expose, enhance and exploit parallelism in the **source code** program to make it explicit in the **machine code** . HP Mgr of **Processor** Architecture Jerry Huck described the four key features of the IA64 combined RISC/CISC architecture as architecture resources, **instruction** format, predication, and speculation.

26/3,K/16 (Item 10 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

02080718 SUPPLIER NUMBER: 19578266 (USE FORMAT 7 OR 9 FOR FULL TEXT)
INTEL'S 64-BIT RISC POINTS THE WAY TO HOW ITS MERCED CHIP MIGHT TURN OUT.
Computergram International, n3199, pCGN07090013
July 9, 1997
ISSN: 0268-716X LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 501 LINE COUNT: 00041

TEXT:
...for a 64-bit RISC-like architecture that can accept multiple operating systems and programs with mixed **instruction** sets - aka Merced - and now the chip-watchers at Electronic Engineering Times have been poring over likely...

...companies, the paper reckons Intel could draw fire from other companies with patents describing ways for a **processor** to execute RISC and CISC **instructions** - such as now defunct Exponential Technology. It even suggests Intel could face new legal challenges from Digital Equipment Corp, which **developed** its own dual-mode **processor** in the 1980s. Meantime, Intel and its IA-64 **instruction** set partner Hewlett- Packard Co have long said that Merced, designed to be compatible with both X86 and PA-RISC **instructions** , will make limited used of VLIW very long- **instruction** -word-type concepts, so it should come as no surprise that Merced is expected to be a more like a traditional RISC device than a native VLIW architecture. Such architectures require new **compilers** to get from source to **object code** ; VLIW basically ties **object code** to one machine. Furthermore, given the high-level security Intel has thrown around Merced to **keep** details from the compatible builders (CI No 3,181), it is by no means clear just how...

...an application written in both iAPX-86 code and IA-64 code can execute on a single **processor** through the use of a mode bit to interpret incoming **instructions** . Electronic Engineering Times says the implication is that only those portions of the application software that can benefit from IA-64 need to be converted, "a trick that should **keep** application code sizes from ballooning." It suggests Intel will opt to gradually convert to IA-64 code...

26/3,K/17 (Item 11 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

02038522 SUPPLIER NUMBER: 19146569 (USE FORMAT 7 OR 9 FOR FULL TEXT)
WHITECROSS KEEPS ITS HEAD BY OFFERING SOMETHING DIFFERENT.
Computergram International, n3104, pCGN02200008
Feb 20, 1997
ISSN: 0268-716X LANGUAGE: English RECORD TYPE: Fulltext
WORD COUNT: 1273 LINE COUNT: 00100

Rudyard Kipling once said, "If you can **keep** your head when all around are losing theirs." With massively parallel processing vendors all around losing their heads, UK company WhiteCross Systems Ltd seems to be **keeping** its own head above water by focusing on a well-defined niche market, data warehousing. The company has recently received a fresh injection of venture capital, won three deals for 100- **processor** **computers** worth more than $1m each, and now has an installed base of 30

(CI No 3,083...

...only if both parties agree the exercise will pay for itself. If the WhiteCross system does not **produce** the agreed benefit, the company commits to refund all fees in full. Bold claims indeed, but WhiteCross...

...based on hardware and software that was designed for transaction processing applications, and based on outdated power- **memory** equations. Holle says massively parallel processing is the natural architecture for large-scale data exploration. Tasks lend themselves to being divided up across many **processors** , so that the time per task can be reduced in proportion to the number of **processors** . Where WhiteCross believes it differs even from the likes of NCR Corp's Teradata is that it is newer to market and has not been handicapped by architecture based on the original IBM personal **computer** . WhiteCross systems are based on the UK-designed Transputer. The WhiteCross system incorporates a grid of interconnecting single **processor** nodes and associated **memory** . Each node communicates with four adjacent nodes so that processing power is not constrained by a single...

...increases in line with the number of nodes added. When an enquiry is made, an on-board **compiler** identifies the tasks that need to be performed and divides them across different nodes. The minimum configuration...

...which, WhiteCross says, brings with it an inherent degradation of performance. WhiteCross applications are compiled directly to **machine code** increasing the speed at which the machine performs tasks by two orders of magnitude. The company says...
...goes through six layers of software architecture, with each layer involving between one and a thousand separate **instructions** , resulting typically in at least 10,000 **instructions** being issued. In contrast, it says, its own proprietary software needs only about 30 **instructions** to complete the same task, which it says at similar **processor** speeds gives an actual speed advantage of 300-fold to WhiteCross. Where the WhiteCross approach really ...ability to let users truly follow what it calls 'train of thought' querying. Hitting any kind of **processor** with totally unstructured types of query is usually a nightmare, and what many database vendors do to minimise the performance hit is to use indexing to define how data tables are **stored** . Indexes are built to reference those columns most likely to be frequently searched. However, this can be very limiting to the end user. Indexing can also **create** additional overhead. WhiteCross instead uses a single image index, which is to index all table data in...

...data loading, since changes to the master index are made on the fly and held in main **memory** , avoiding the need for disk access. So the answer from WhiteCross is not just to throw raw...

26/3,K/22 (Item 16 from file: 275)

01599347 SUPPLIER NUMBER: 13762064 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Operating environments. (Solutions) (question-and-answer)
Rosenberg, Allan; Rubenking, Neil J.; Lichtensteiger, Manfred
PC Magazine, v12, n11, p369(2)
June 15, 1993
ISSN: 0888-8507 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT
WORD COUNT: 952 LINE COUNT: 00070

... change, you'll be prompted to restart Windows--don't do it! Instead, exit Windows entirely.
Now **use** PC Magazine's DEFRAGR (February 23, 1993) or a similar program to defragment the target disk, and...

...number of ways you can do this; one method is to create a document in your word **processor** and **keep** doubling its size **using** cut-and-paste.

Or you can **use** the tiny Pascal program shown in Figure 2 to eat up disk space in increments of 1K. (You can download the **source code** and **executable** files from PC MagNet, archived as EATDIS.ZIP.) The program will compile under either the DOS or Windows versions of Borland's Pascal **compiler** . Invoke it by passing the number of kilobytes to grab and the directory in which to place its **temporary file** . If the parameters are valid, the program creates EATDISK.directory with the specified size; if not, it...

...cluster size of your hard disk, typically 2K. For example, to consume 4 megabytes on drive D:, **use** the command EATDISK 4096 D:\.
When the space remaining on the target drive has been reduced to...

26/3,K/25 (Item 19 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01562950 SUPPLIER NUMBER: 14466824
DSP use poses programming challenges. (Embedded Systems: Part 4)
Hosking, Rodger
Electronic Engineering Times, n766, p59(1)
Oct 4, 1993
ISSN: 0192-1541 LANGUAGE: ENGLISH RECORD TYPE: ABSTRACT

ABSTRACT: Program **developers** should be aware of the unique programming differences between programming digital signal **processors** (DSP) and general-purpose controllers because DSP application features operate in conflict with each other. Key programming issues include I/O library support, differences between native and cross- **compiler** techniques, **memory** and **linker** models. Null environment C cross **compilers** for DSPs execute **object code** on the DSP central **processor** unit (CPU) and not on the host CPU. Simulators emulate DSP operations and are useful for designing...

...can exist between the DSP target and host when an operating system operates on the DSP. C **compilers** for DSP applications can **produce** an assembly-language output file, while other **compilers** will **produce** an object file from the C **source code** . An advantage of assembly-language C **compilers** is the ability to inspect and modify the assembly code as part of the debugging process.

26/3,K/26 (Item 20 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01520901 SUPPLIER NUMBER: 12337903 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Eiffel/S - Eiffel for the rest of us. (Software Review) (the Eiffel programming language compiler for MS-DOS from Germany's Sig Computer GmbH) (Evaluation)
Johnson, Paul
EXE, v7, n1, p66(4)
June, 1992
DOCUMENT TYPE: Evaluation ISSN: 0268-6872 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2998 LINE COUNT: 00228

...ABSTRACT: that implements an algorithm. The compiler produces C source code rather than machine code. The C code **generated** is clear and documented. A C interface enables interfacing with routines and libraries written in other programming languages. Drawbacks include the lack of a C **compiler** , non-conformance to the Eiffel standard in several ways, lack of a **development** environment, a proprietary **memory** extender that is compatible with little other software, difficult debugging and awkward error management.

26/3,K/35 (Item 29 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01249754 SUPPLIER NUMBER: 06736839 (USE FORMAT 7 OR 9 FOR FULL TEXT)
88000 designed for use with optimizing compilers. (Motorola 88000 chip)
Falk, Howard
Computer Design, v27, n9, p30(2)
May 1, 1988
ISSN: 0010-4566 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 1759 LINE COUNT: 00144

ABSTRACT: Motorola's new 88000 reduced- **instruction** -set computer (RISC) chip set was designed to be used with optimizing compilers in order to maximize...

...implementation provided the best operation. Optimizations include value tracking, which reduces value access times when values are **stored** and tracked in one of the CPU's 31 general purpose registers, and the mapping of high...

...tools are available, including UNIX C, Pascal, and Fortran functions; a symbolic debugger; a linker to combine **object** - **code** modules; and a driver routine for processing **source** **code** .

26/3,K/36 (Item 30 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01237139 SUPPLIER NUMBER: 06172706 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Compiler flexibility is particularly important in hybrid environments. (mixed-language programming)
Jenkins, Avery
PC Week, v5, n2, p103(1)
Jan 12, 1988
ISSN: 0740-1604 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 562 LINE COUNT: 00045

...ABSTRACT: respond. Many language developers are dividing their compilers into at least two pieces in an attempt to **keep** up with changing standards, which allows them to modify the compilers more easily for different languages or hardware platforms. The first part translates the **source** **code** into an intermediate language that is then translated by the compiler's second part, or back end, into the program's **object** **code** .

26/3,K/47 (Item 4 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

03577119 Supplier Number: 47404318 (USE FORMAT 7 FOR FULLTEXT)
REPORT HAS IBM WORKING ON VERY LONG INSTRUCTION WORD CHIP TO RUN JAVA AND POWERPC OBJECT CODE
Computergram International, n3166, pN/A
May 22, 1997
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 325

(USE FORMAT 7 FOR FULLTEXT)
TEXT:
...Java and PowerPC object code that could be used in boards for set-top devices and Network **Computers** . The chips would supposedly be different from other Java chips that execute Java code in hardware, such as Sun Microsystems Inc's picoJava, in its use of VLIW Very Long Word **Instruction**

Word techniques. A VLIW chip, by definition an **instruction** -level parallel architecture, would enable multiple **instructions** , each with many bits - hence long word - to run in one clock cycle on multiple execution units...

...design than either CISC or RISC, because complexity is transferred to the software, VLIW requires a clever **compiler** to schedule the **instructions** for processing across the execution units. No existing **object code** can run on a VLIW chip without a specific **instruction** set converter. This explains why previous VLIW efforts have largely failed to get off the ground although applications arebeing **developed** for use with the VLIW properties of the specialised Philips TriMedia and Chromatic Research Inc Mpact multimedia **processors** . A paper posted on IBM's TJ Watson Research Center web server in Yorktown Heights in New...

...Center has been working on VLIW techniques since 1986. A VLIW software translator called Daisy - Dynamically Architected **Instruction** Set from Yorktown - will reportedly convert Java code and PowerPC **machine code** that the VLIW **processor** can execute. It eventually plans to **develop** compatibility for Intel iAPX-86 and S/390 **processors** . The chip could be used as part of a single-board network **computer** that would include the chip, **cache memory** , RAM and boot ROM, and a PowerPC-to-PCI bus bridge to various connections on the board...

26/3,K/48 (Item 5 from file: 636)

03464731 Supplier Number: 47142006 (USE FORMAT 7 FOR FULLTEXT)
WHITECROSS KEEPS ITS HEAD BY OFFERING SOMETHING DIFFERENT
Wallen, Joanne
Computergram International, n3104, pN/A
Feb 20, 1997
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 1200

Rudyard Kipling once said, "If you can **keep** your head when all around are losing theirs." With massively parallel processing vendors all around losing their heads, UK company WhiteCross Systems Ltd seems to be **keeping** its own head above water by focusing on a well-defined niche market, data warehousing. The company has recently received a fresh injection of venture capital, won three deals for 100- **processor computers** worth more than $1m each, and now has an installed base of 30 (CI No 3,083...

...only if both parties agree the exercise will pay for itself. If the WhiteCross system does not **produce** the agreed benefit, the company commits to refund all fees in full. Bold claims indeed, but WhiteCross... ...based on hardware and software that was designed for transaction processing applications, and based on outdated power- **memory** equations. Holle says massively parallel processing is the natural architecture for large-scale data exploration. Tasks lend themselves to being divided up across many **processors** , so that the time per task can be reduced in proportion to the number of **processors** . Where WhiteCross believes it differs even from the likes of NCR Corp's Teradata is that it is newer to market and has not been handicapped by architecture based on the original IBM personal **computer** . WhiteCross systems are based on the UK-designed Transputer. The WhiteCross system incorporates a grid of interconnecting single **processor** nodes and associated **memory** . Each node communicates with four adjacent nodes so that processing power is not constrained by a single...

...increases in line with the number of nodes added. When an enquiry is made, an on-board **compiler** identifies the tasks that need to be performed and divides them across different nodes. The minimum configuration...

...which, WhiteCross says, brings with it an inherent degradation of performance. WhiteCross applications are compiled directly to **machine code** increasing the speed at which the machine performs tasks by two orders of magnitude. The company says...
...goes through six layers of software architecture, with each layer involving between one and a thousand separate **instructions** , resulting typically in at least 10,000 **instructions** being issued. In contrast, it says, its own proprietary software needs only about 30 **instructions** to complete the same task, which it says at similar **processor** speeds gives an actual speed advantage of 300-fold to WhiteCross. Where the WhiteCross approach really ...ability to let users truly follow what it calls 'train of thought' querying. Hitting any kind of **processor** with totally unstructured types of query is usually a nightmare, and what many database vendors do to minimise the performance hit is to use indexing to define how data tables are **stored** . Indexes are built to reference those columns most likely to be frequently searched. However, this can be very limiting to the end user. Indexing can also **create** additional overhead. WhiteCross instead uses a single image index, which is to index all table data in...

...data loading, since changes to the master index are made on the fly and held in main **memory** , avoiding the need for disk access. So the answer from WhiteCross is not just to throw raw...

26/3,K/49 (Item 6 from file: 636)
DIALOG(R)File 636:Gale Group Newsletter DB(TM)

02404347 Supplier Number: 44761701 (USE FORMAT 7 FOR FULLTEXT)
TO OPTIMISE OR NOT TO OPTIMISE: THE QUESTION POSED BY EVOLVING POWERPC FAMILY
Computergram International, n2436, pN/A
June 15, 1994
Language: English Record Type: Fulltext
Document Type: Newswire; Trade
Word Count: 1292

... method of using dynamically linkable libraries, which can be swapped in at run-time depending on the **processor** in use. If they are restricted to specialist graphical library use, there should not be too many problems. However, other informed IBMers suggest that these **instructions** will become more pervasive over time as the focus switches to floating point usage in general applications...

...what is the answer? Motorola's Phillip suggests one solution may be smart installers. In Motorola's **compilers** , like many others, switches abound. **Developers** can compile for particular targets, and for particular chips. They also allow for the production of multiple sets of **object code** . So, an installation CD-ROM could hold multiple **copies** of (parts of?) the application, together with a smart installer that could read the system configuration, **processor** -type and so forth, and **copy** the most suitable code across. It is an evolution of the fat binary idea. Will it catch...

26/3,K/53 (Item 2 from file: 813)
DIALOG(R)File 813:PR Newswire

1131187 LAM035
Toshiba Announces Its First TX19 Microprocessor Device First Standard Product to Implement the MIPS16 Instruction Set Architecture

DATE: July 28, 1997 08:08 EDT WORD COUNT: 846

...MIPS-16 ISA as well

as upward object code compatible with the TX39 family (Selected MIPS ASE **instructions** are not supported).
The TX19 core features a five-stage pipeline, four cycle multiply accumulate (MAC), as well as 4KB **instruction** **cache** , 1KB data **cache**
and on-chip debug support.
Toshiba and third party suppliers support multiple **development** environments to provide seamless **development** between Toshiba's MIPS
RISC TX cores. **Development** environments and operating systems are
available from the following third party suppliers :
-- Cygnus Solutions (GNUPro tool suite)
-- Toshiba (C/C++ **compiler** , **assembler** , **linker** , debugger, in
circuit emulator)
-- Hewlett-Packard (Debugger probe)
-- Additional suppliers to be announced

Pricing and Availability:

Samples...

26/3,K/54 (Item 3 from file: 813)
DIALOG(R)File 813:PR Newswire

1131186 LAM036
Toshiba Announces Architecture for New Low-Power TX19 Microprocessor Family

DATE: July 28, 1997 08:08 EDT WORD COUNT: 1,077

...based RISC designs.

Advanced Features:
Improved code efficiency by nearly 40 percent versus current 32-bit RISC **processors** .
Low-power/high-performance ratio. The TX19 core has 1,000 MIPS/W (3.0V, 20MHz).
Optimized...

...The
interrupt response time of the TX19 is one eighth of that of current 32-bit RISC **Processors** .
The TX19 is compatible with MIPS-I, MIPS-II and MIPS-16 **ISA** as well as
upward **object** **code** compatible with the TX39 family (Selected MIPS ASE
instructions are not supported).
The TX19 core features a five-stage pipeline, four cycle multiply accumulate (MAC), as well as 4KB **instruction** **cache** , 1KB data **cache** and
on-chip debug support.
Toshiba and third party suppliers support multiple **development** environments to provide seamless **development** between Toshiba's MIPS
RISC TX cores. **Development** environments and operating systems are
available from the following third party suppliers .
-- Cygnus Solutions (GNUPro tool suite)
-- Toshiba (C/C++ **compiler** , **assembler** , **linker** , debugger, in circuit
emulator)
-- Hewlett-Packard (Debugger probe)
-- Additional suppliers to be announced

The TX19 is a...